

仮想レオロジー物体における 視触覚提示に関する研究

(研究課題番号 14205039)

平成14年度～平成16年度 科学研究費補助金
基盤研究(A)(2) 研究成果報告書

平成17年3月

研究代表者 立命館大学 ロボティクス学科
教授

平井 慎一

まえがき

本研究の目的は、実世界に数多く存在する柔軟なレオロジー物体と、同等の視触覚情報を人間に提示するために必要な基盤技術を確立することである。レオロジー物体とは、レオロジー的特性を有する三次元物体であり、血管や筋肉などの生体組織や食品がその代表的な例である。レオロジー物体に力を作用させると、物体の三次元形状が大きく変化し、ヒステリシスや非線形性など複雑な力学的特性を示す。仮想レオロジー物体とは、コンピュータ内に構築され、人間が与える力や運動に対して、実際と同様の視触覚情報を提示するコンピュータシステムである。

近年、医療分野においては、遠隔手術システムに代表されるロボットシステムの導入が試みられている。このようなロボットシステムを用いる手術は、従来の手術と異なり、機器を操作する手元と、手術を実行する箇所とが異なるため、あらかじめ手術用シミュレータを用いて、遠隔手術のトレーニングやシミュレーションを重ねることが必要である。血管、筋肉、内臓など、生体組織の多くは、レオロジー的性質を有する材質から構成されており、複雑な三次元変形特性を示す。したがって、これらレオロジー特性を有する三次元物体の変形をシミュレートし、その変形や力感覚をオペレータに提示するシステムの開発が望まれている。しかしながら、1) 実レオロジー物体の特性を同定し、仮想レオロジー物体を構築する手法が確立していない、2) 仮想レオロジー物体の変形プロセスの計算量が多く、実時間での演算が困難であるという障害のため、レオロジー特性を有する物体をオペレータに提示するシステム、すなわち本研究で目的とする仮想レオロジー物体は実現されていない。レオロジー物体は様々な特性を持っているので、仮想レオロジー物体を構築するためには、実レオロジー物体の変形プロセスを計測し、レオロジー特性を規定するモデルパラメータを同定することが必要である。レオロジーの分野では、一次元の変形から材料の特性を同定する手法が確立している。しかしながら、実レオロジー物体の変形は三次元であり、しかも相互に干渉が強い。したがって、一次元の変形計測のみでは、レオロジー物体の特性を同定することはできない。レオロジー物体の特性を同定するためには、レオロジー物体の三次元変形プロセスを計測するとともに、計測値からモデルパラメータを同定する手法を確立することが必要である。また、仮想レオロジー物体を実現するためには、物体の変形と力をリアルタイムで計算し、計

算結果を視触覚ディスプレイに提示する必要がある。しかしながら，仮想物体の構築で用いられる手法の多くは，個々の変形要素における計算が単純である一方，全体の計算量が極めて多い。結果として，仮想レオロジー物体の変形と力を，リアルタイムでは計算できない。仮想レオロジー物体の変形と力をリアルタイムで提示するためには，高速演算法を開発することが望まれる。以上の経過を経て，研究代表者は，仮想レオロジー物体における視触覚提示に関する研究の重要性を認識し，本研究課題を提案した。

第1章では，レオロジー変形のパーティクルベースモデリングについて述べる。物体変形のモデリング手法は大きく，パーティクルベースモデルと連続体モデルに分類できる。パーティクルベースモデルは，大変形に対応でき，連続体モデルと比較すると計算時間が2桁以上短いため，変形提示に適していると考えられる。ただし，従来の研究は弾性体や塑性体を対象としており，従来手法を直接適用するとレオロジー的な変形を安定に計算できない。そこで本章では，レオロジー変形を安定に計算できるパーティクルベースモデリングの手法を明らかにする。(成果 [13, 17, 19, 20, 21])

第2章では，FPGAによる仮想レオロジー物体のリアルタイム変形計算について述べる。前章で述べたパーティクルベースモデルは，連続体モデルより計算時間が短い，触覚提示に必要なリアルタイム性は全く不十分である。力覚提示を伴う場合には，1000Hz以上のサンプリングレートでの計算が必要となり，また安定で高品質な力覚提示を実現するためには，1000Hzよりも高いサンプリングレートでの計算が必要であることが報告されている。したがって，このようなサンプリングレートに対応できる変形計算が必要である。パーティクルベースモデルにおける物体変形の計算では，個々のパーティクルにおける計算は単純であるが，パーティクルがモデル内に多数あるため，結果として全体の計算量が増えている。このような変形計算においては，FPGA (Field Programmable Gate Array) に計算回路を構成することにより，計算の高速化が期待できる。そこで本章では，パーティクルベースモデルにおける変形計算をFPGAに回路として実装し，仮想レオロジー物体のリアルタイム変形計算を実現する。(成果 [1, 2, 16])

第3章では，視触覚提示のための仮想レオロジー物体のキャリブレーションについて述べる。パーティクルベースモデルは，大変形に対応でき，連続体モデルと比較すると計算時間が短いという長所がある反面，モデル内の多数の力学パラメータが複雑に関連しており，モデル同定が困難であるという欠点を有する。連続体モデルでは，ヤング率や体積弾性率あるいは粘性率という，力学的に明快なパラメータを用いているのとは，対照的である。そこで本章では，レオロジー物体の三次元変形形状と物体に作用する力を計測するシステムを構築するとともに，randomized algorithmを用いて，パーティクルベースモデルの実際のレオロジー物体の計測値から，力学

パラメータを同定する手法を確立する．この randomized algorithm は，もともと径路計画で開発された手法である．本章では，物体の計測値から力学パラメータを同定する過程を，パラメータ空間内の径路計画とみなすことにより，仮想レオロジー物体の力学パラメータを同定する．(成果 [3, 4, 5, 12, 14, 15, 18])

第4章では，連続体力学に基づく物体変形のパーティクルベースモデリングを述べる．前述のように，パーティクルベースモデルと連続体モデルは，相反する特徴を持つ．そこで，連続体力学における概念を基礎として，パーティクルベースモデルを構築することを試みた．実際のレオロジー物体の多くは，非圧縮性に近い特性を示す．しかしながら，連続体力学における変形計算では，ポワソン比が0.5という非圧縮性の場合には，計算が発散し，安定な計算ができない．そこで，CSM (Constraint Stabilization Method) を用いて，非圧縮性をモデリングし，非圧縮性変形を安定に計算する手法を提案する．(成果 [6])

第5章では，仮想レオロジー物体構築のための一般化フォクトモデルについて述べる．実際のレオロジー物体は複雑な非線形特性を示す．このような非線形特性を，非線形要素を用いてモデリングすることは，モデル同定が困難であること，計算時間が増大することを勘案すると，適切ではない．そこで本章では，仮想レオロジー物体のモデリングに，一般化フォクトモデルを導入することを試みる．

以上のように本研究課題では，1) パーティクルベースモデリングによるレオロジー物体の大変形モデリング技術の確立，2) FPGA(Field Programmable Gate Array) によるリアルタイム変形計算，3) レンジファインダと分布圧センサを用いたリアルタイム変形計測法の確立，4) Randomized Algorithm による仮想レオロジー物体のモデルパラメータ同定技術という成果を得た．得られた成果により，厚さが10分の1に圧縮される生地の捏ね過程のシミュレーション，1000点規模のパーティクルベースモデルのビデオフレームレート実行，変形誤差10%以内のモデル同定が可能になっている．また，本研究課題の成果は，人工筋肉アクチュエータのモデリング [7, 8]，軟骨を含む人体の関節のモデリング [9]，間接同時位置決め [10, 11] に応用されている．特に，人工筋肉アクチュエータは，レオロジー的な変形特性を示すことが多く，本研究課題の成果を有効に適用できる．また，人体の関節や生体組織など，レオロジー的特性を示し，そのモデル同定が困難な対象は多い．生体組織のモデリングとモデル同定は，今後の課題の一つである．

本研究課題を遂行するなかで，手術や触診のシミュレーションや食感の分析，触覚の提示においては，物体形状や表面圧力のみならず，柔軟物内部の変形や力学量の分布を同定し，それに基づいた柔軟物モデリングが必須であるとの認識に至った．すなわち，柔軟物内部の挙動を解明するためには，1) マイクロメカニカルセンサを内部に埋め込み，力や加速度を計測することと，2) CT や MRI に代表される3次元

イメージングを用いて内部の変形を計測すること，3) これらのセンシングを融合することが必要であるとの認識に至った．すなわち，本研究課題を含む従来の研究では，表面に現れる変形特性を表現することが中心的な課題であり，内部の挙動や力学量の分布に関しては，センシングの手法が限られていることもあり，未開拓の部分が多い．これは，次の課題としたい．

目次

第1章	レオロジー変形のパーティクルベースモデリング	1
1.1	緒言	1
1.2	仮想レオロジー物体のモデリング	2
1.2.1	三要素モデル	2
1.2.2	トラス構造	4
1.2.3	位相保持	4
1.2.4	体積効果のモデリング	7
1.2.5	同一モデルによる弾性，粘塑性，レオロジー物体の表現	9
1.3	物体間の相互作用のモデリング	9
1.3.1	従来のペナルティ法	11
1.3.2	ペナルティ時間積分法	11
1.3.3	動摩擦のモデリング	13
1.4	シミュレーション	13
1.5	結言	13
1.6	資料	16
第2章	FPGAによる仮想レオロジー物体のリアルタイム変形計算	69
2.1	緒言	69
2.2	並列計算向きの変形計算アルゴリズム	70
2.2.1	レオロジー物体の定義	70
2.2.2	三要素モデル	71
2.2.3	一般化フォークトモデル	73
2.3	変形計算アルゴリズムのFPGA実相	74
2.3.1	仮想レオロジー物体の変形モデル	74
2.3.2	計算回路の構成	75
2.3.3	誤差の抑制	79
2.3.4	FPGA実装	82
2.3.5	実現可能な処理能力	83
2.4	結言	86

2.5	資料	87
第3章	視触覚提示のための仮想レオロジー物体のキャリブレーション	91
3.1	緒言	91
3.2	Randomized Algorithm によるレオロジー変形特性の同定	92
3.3	Randomized Algorithm によるキャリブレーション	100
3.4	Modified Randomized Algorithm によるキャリブレーション	108
3.5	体積効果を含むラティスモデルのキャリブレーション	116
3.6	トラスモデルのキャリブレーション	124
第4章	連続体力学に基づく物体変形のパーティクルベースモデリング	132
4.1	緒言	132
4.2	剪断モデル	132
4.2.1	二次元物体における剪断モデル	132
4.2.2	二質点にかかる反作用の導出	135
4.2.3	三次元剪断モデルにおける剪断力	137
4.3	体積モデル	140
4.3.1	二次元物体における体積変形	140
4.3.2	三次元物体における体積変形	141
4.4	CSM による非圧縮性のモデリング	142
4.4.1	面積一定制約	143
4.4.2	体積一定制約	145
4.4.3	パラメータの設定	149
4.5	シミュレーション	150
4.5.1	四角形剪断モデルによる物体変形シミュレーション	150
4.5.2	三角形剪断モデルおよび体積モデルによる弾性物体，粘塑性物体，レオロジー物体の変形シミュレーション	150
4.6	結言	152
4.7	資料	153
第5章	仮想レオロジー物体構築のための一般化フォークトモデル	174
5.1	緒言	174
5.2	一般化フォークトモデル	174
5.2.1	一般化フォークトモデル	175
5.2.2	内部摩擦考慮型粘性モデル	176
5.3	変形シミュレーション	177

5.3.1	二要素一般化フォークトモデルの変形シミュレーション	177
5.3.2	三要素一般化フォークトモデルの変形シミュレーション	179
5.3.3	内部摩擦考慮型三要素モデルの変形シミュレーション	179
5.4	力学解析とパラメータ推定	181
5.4.1	荷重中に一般化フォークトモデルの質点に作用する力	183
5.4.2	二要素一般化フォークトモデルの解析	184
5.4.3	二要素一般化フォークトモデルの粘弾性パラメータ抽出	185
5.5	実験結果	189
5.5.1	低反発ウレタンスポンジの変形実験	189
5.5.2	実験結果とシミュレーションの比較	192
	参考文献	195
	成果	198

第1章 レオロジー変形のパーティクルベースモデリング

1.1 緒言

バーチャルリアリティの分野では、従来より、弾性物体や塑性物体のモデリングに関する研究が多くなされてきた。しかし、現実世界には、弾性物体と塑性物体の中間的な性質を示すレオロジー物体が数多く存在する。レオロジー物体の例としては、パン生地やゼリー等の食品類、臓器や血管などの生体組織、粘土が挙げられる。これらを仮想空間に構築することで、手術、陶芸、食品加工などの幅広い分野におけるシミュレータへの応用が可能であると考えられる。しかしながら、レオロジー物体のモデリングに関する研究は、これまでほとんど行われていない。本章では、仮想レオロジー物体を構築することを目的として、レオロジー物体のモデリングを行う。

レオロジー物体の性質 図 1.1-(a) に示す初期形状を有する物体に外力を作用させると、図 1.1-(b) に示すように変形すると仮定する。弾性物体では、図 1.1-(c) に示すように、外力を解放したときの形状が初期形状に一致する。外力を作用させた形状と外力解放後の形状との差を、戻り変位、初期形状と外力解放後の形状との差を残留変位とよぶ。弾性物体では、戻り変位があり、残留変位はない。粘塑性物体では、図 1.1-(d) に示すように、外力を解放したときの形状が変形形状に一致する。すなわち粘塑性物体では残留変位があり、戻り変位はない。本報告では、図 1.1-(e) に示すように戻り変位と残留変位の両方を有する物体をレオロジー物体と定義する。

先行研究 現在、対話操作に適した実時間変形計算の可能な仮想弾性物体のモデリングに関する研究が精力的に行なわれている [5, 2]。しかし、これらの研究で取り扱う物体は弾性物体に限られている。一方、材料学の分野においては、古くからレオロジー物体の特性に関する研究がなされてきた [18, 14]。また、近年では分子レベルにおけるレオロジー物体の動力学シミュレータの開発が進められている [11]。しかし、これらは材料特性を対象としており、三次元の変形形状は扱われていない。レオロジー物体の変形モデルとしては、徳本らによるモデルが挙げられる [24]。しか

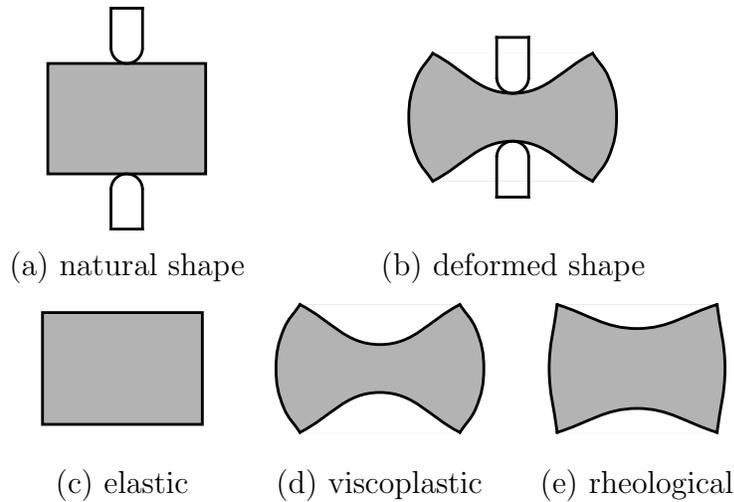


図 1.1: 弾性物体，粘塑性物体，レオロジー物体

し，このモデルでは変形形状が実際のレオロジー物体と大きく異なり，また，物体の大変形の際に変形計算が不安定となる．Terzopoulos らは，並進回転からなる剛体運動と物体変形を組み合わせることによりレオロジー物体の構築を行っている [22]．これに対し，本報告では，実装が容易で高速化のための並列計算に適した，パーティクルベースモデルに基づくレオロジー物体のモデリング手法を提案する．CG の分野では，Reeves が提案したパーティクルベースモデル [16] が，変形モデリングに導入されている [5, 2]．しかし，それらの研究は弾性的な性質を表現するものであり粘塑性を含んだ変形モデルはまだ確立されていない．

1.2 仮想レオロジー物体のモデリング

本節では，残留変位や戻り変位を持つ力学要素を用いることで，仮想レオロジー物体のモデリングを行う．さらに位相保持および体積効果を追加することで，実レオロジー物体に近い変形挙動を実現する．

1.2.1 三要素モデル

パン生地などの特性を表現するためには戻り変位量，残留変位量，戻り変位速度を独立に指定する必要がある．これらの特性を独立に指定できる最小の力学モデルが三要素モデルである．このモデルは図 1.2 に示すように，戻り変位を生じるフォークトモデルと残留変位を生じる独立したダンパの直列によって構成される．本研究で

はレオロジー的性質を表現するための一次元要素としてこの三要素モデルを用いる。

三要素モデルの長さおよび三要素モデルのフォークト部の長さをそれぞれ l , l_{voigt} とし, フォークト部の粘性係数, 弾性係数および自然長を k_1 , c_1 , L , 独立ダンパ部の粘性係数を c_2 とする. 独立ダンパ部の長さ l_{damper} は, 次式で表される.

$$l_{damper} = l - l_{voigt}. \quad (1.1)$$

フォークト部に作用する力 f および独立ダンパ部に作用する力 f_{damper} は,

$$f = -k_1(l_{voigt} - L) - c_1(\dot{l}_{voigt}), \quad (1.2)$$

$$\begin{aligned} f_{damper} &= -c_2(\dot{l}_{damper}) \\ &= -c_2(\dot{l} - \dot{l}_{voigt}) \end{aligned} \quad (1.3)$$

となる. ここで, フォークト部に作用する力および独立ダンパ部にかかる力は等しいことから, (1.2), (1.3) 式より, フォークト部の長さの変化に関する式が導かれる.

$$\dot{l}_{voigt} = \frac{-k_1(l_{voigt} - L) + c_2\dot{l}}{c_1 + c_2}. \quad (1.4)$$

変形計算中に三要素モデルのパラメータが変化しないものと仮定し, 定数 $A = -k_1/(c_1 + c_2)$, $B = c_2/(c_1 + c_2)$ を導入すると, (1.4) 式は次式の形となる.

$$\dot{l}_{voigt} = A(l_{voigt} - L) + B\dot{l}. \quad (1.5)$$

質点 P_i を始点とする三要素モデルの集合を R_i , 質点 P_i を終点とする三要素モデルの集合を S_i で表す. 稜線に向きを付け, 第 k 稜線の始点から終点に向かう単位ベクトルを e_k と表す. このとき, 集合 R_i に含まれる三要素モデル E_k が, 質点 P_i に加える力は $f_k e_k$ に一致する. また, 集合 S_i に含まれる三要素モデル E_k が, 質点 P_i に加える力は $-f_k e_k$ に一致する. したがって, 質点 P_i の運動方程式は,

$$m_i \dot{v}_i = \sum_{k \in R_i} f_k e_k - \sum_{k \in S_i} f_k e_k + F_i^{ext} \quad (1.6)$$

と表される. ここで F_i^{ext} は, 質点 P_i に作用する外力である. 結局, 物体モデルの運動方程式は, (1.5)(1.6) 式で与えられる. 仮想レオロジー物体の変形は, 運動方程式を数値的に解くことにより, 計算することができる. 本研究ではルンゲクッタ法を用いて計算を行う.

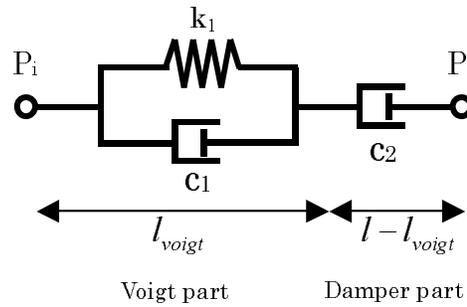


図 1.2: 三要素モデル

1.2.2 トラス構造

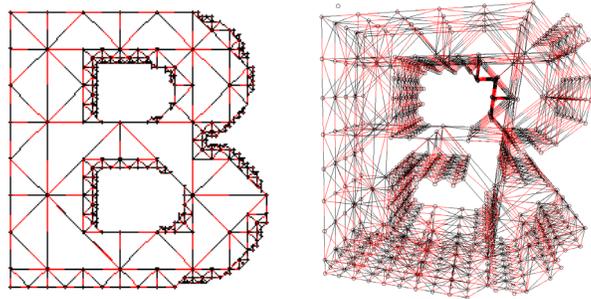
一次元要素である三要素モデルを格子状に組み合わせることによって、二次元物体および三次元物体を表現することが可能である。本研究ではグラフィック表示との相性に優れる三角面および四面体の集合によって、任意の形状を持つ二次元物体および三次元物体を構築する。仮想物体の変形計算に用いる要素として、質点、稜線、三角面および四面体を用意する。稜線、三角面、四面体は、向きを有する要素として扱う。二次元形状を現すデータとして三角面の集合、三角面を構成する各稜線と対向する頂点の組および稜線の向きを保持する。三次元形状を現すデータとして四面体の集合、四面体の各三角面と対向する頂点の組および三角面の向きを保持する。

図 1.3 に、トラス構造を用いて物体の形状を表現した例を示す。図 1.3-(a) は二次元形状、図 1.3-(b) は三次元形状の表現例である。図に示すように、大きさの異なる三角形あるいは四面体を用いて形状を表現する。図 1.3 の例では稜線上に質点が生成されているため、物体変形時にクラックと呼ばれる物体内部の亀裂を生じることがある。しかし、自然長を 0 としたフォークトモデルで質点と稜線を接続することで、クラックの発生は防止することができる。

1.2.3 位相保持

一次元要素のトラス構造によるモデルでは、大変形の際に位相が崩れ、三角形や四面体などが反転することがある。その場合、元の形状に戻ろうとする復元力が失われ、三角形や四面体が鏡像の状態で安定するため、適切な戻り変位が得られない。そこで本節では、トラス構造モデルにおける位相的な接続関係を保つ手法を検討する。

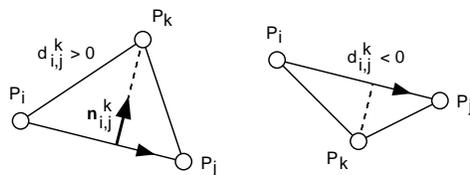
二次元物体モデルにおいて位相的な接続関係が崩れるのは、図 1.4-(a),(b) に示すような鏡像どうしを区別できないことに起因する。そこで、鏡像どうしが区別でき



(a) 2D object

(b) 3D object

図 1.3: トラス構造による形状表現の例



(a) positive

(b) negative

図 1.4: 頂点と稜線の符号付き距離

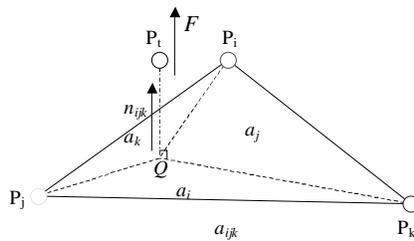


図 1.5: 三角面に作用する力の配分

るように，個々の稜線に向きを付ける．稜線に向きが定義されているので，図 1.4 に示すように，三角要素の格子点 P_k と稜線 P_iP_j との間に，符号付距離を定義することができる．稜線 P_iP_j に垂直で，自然状態における P_k を向く単位ベクトルを \mathbf{n}_{ij}^k ，稜線 P_iP_j と格子点 P_k との符号付き距離を d_{ij}^k とする．位相的な接続関係を保つためには，符号付き距離 d_{ij}^k の値が小さくなったとき，質点 P_k に稜線から離れる方向の力を作用させる．本報告では，このような力が仮想的なフォークト要素により生成されると考え，以下に示す人工的な力を導入する．

$$\mathbf{f}_{ij}^k = \begin{cases} \mathbf{0} & (d_{ij}^k > \epsilon) \\ \{-K(d_{ij}^k - \epsilon) - Cd_{ij}^k\}\mathbf{n}_{ij}^k & (d_{ij}^k \leq \epsilon) \end{cases} \quad (1.7)$$

ここで， K, C はフォークト部の弾性係数，粘性係数， ϵ は微小な正の定数である．符号付き距離 d_{ij}^k が閾値 ϵ より小さくなったとき，仮想的なフォークト要素が生成する力により，質点 P_k は稜線 P_iP_j より離れる方向に動く．

位相保持フォークト要素が発生する力 \mathbf{f}_{ij}^k は質点 P_k に与えられる．作用反作用の法則により，稜線にも逆向きの力を与える．稜線は質量をもたないので，稜線にかかる力を稜線を構成する両端の質点 P_i と P_j に分配する必要がある．そこで，質点 P_k から稜線 P_iP_j へ下ろした垂線による稜線の内分点を Q とし， Q 点によって P_iP_j を内分する比率 r_{ij}^k を定義する．そして質点 P_i および P_j に対して，それぞれ $r_{ij}^k \mathbf{f}_{ij}^k$ ， $(1 - r_{ij}^k) \mathbf{f}_{ij}^k$ の力を与えることにより， Q 点周りのモーメントが 0 となるように力が分配される．しかし， r_{ij}^k は負の値や，1 を超える値になることがあり，三角面が極度に横に崩れた形状となった場合に，各質点に対して極端に大きな力が作用し，変形計算が不安定となることがある．そこで以下の拘束を加え，比率 r_{ij}^k の代わりに比率 R_{ij}^k を用いて力を分配する．

$$R_{ij}^k = \begin{cases} 0 & (r_{ij}^k < 0) \\ r_{ij}^k & (0 \leq r_{ij}^k \leq 1) \\ 1 & (r_{ij}^k > 1) \end{cases} \quad (1.8)$$

なお，三次元物体モデルにおいては，四面体要素の頂点と対向する三角形との間で，同様の符号付き距離が定義でき，上述の議論が適用できる．また反力を分配する方法も同様に三角形の面積比率を使い求めることができる．この時，二次元の場合と同様，四面体が極度に横方向に崩れた場合に変形計算が不安定にならないように面積比率に拘束を加える．図 1.5 に示すように，質点 P_t が三角形 $P_iP_jP_k$ と対向しているとする．質点 P_t の三角形 $P_iP_jP_k$ への足が三角形を内分する点を Q とする．三角形 $P_iP_jP_k$ の面積を a_{ijk} とし，三角形 QP_jP_k ，三角形 QP_kP_i ，三角形 QP_iP_j のそれぞれの符号付面積を a_i, a_j, a_k とする．各質点に対応する面積比率を r_i, r_j, r_k とする．

まず，負の値を切り捨てた面積比率 r_i^{pos} を算出する．

$$r_i^{pos} = \begin{cases} 0 & (r_i \leq 0) \\ r_i & (r_i > 0) \end{cases} . \quad (1.9)$$

同様に面積比率 r_j^{pos} , r_k^{pos} を求める．さらに，比率の合計が1となるよう，拘束後の面積比率 R_i を決定する．

$$R_i = \frac{r_i}{r_i^{pos} + r_j^{pos} + r_k^{pos}} . \quad (1.10)$$

同様に R_j , R_k を求める．位相保持フォークト要素により，質点 P_t に作用する力を f_{ijk}^t とする．このとき，質点 P_i , P_j , P_k に，それぞれ $R_i f_{ijk}^t$, $R_j f_{ijk}^t$, $R_k f_{ijk}^t$ の力を分配する．

また位相的な接続関係が崩れる原因で，三要素モデル特有の問題として，フォークト部と独立ダンパ部の長さの比率がある．三要素モデルの長さを l とすると，フォークト部の長さ l_{voigt} は次式に示す条件を満たさなくてはならない．

$$0 \leq l_{voigt} \leq l . \quad (1.11)$$

しかし， l_{voigt} は計算過程で (1.11) 式で示された範囲を超える可能性がある．そこで0から1までの値をとる定数 a_{\min} , a_{\max} をおき， l_{voigt} に以下の制約条件を課す．

$$la_{\min} \leq l_{voigt} \leq la_{\max} . \quad (1.12)$$

計算過程で l_{voigt} の値が la_{\min} を下回った時には l_{voigt} に la_{\min} の値を代入し， l_{voigt} の値が la_{\max} の値を上回った場合には， l_{voigt} に la_{\max} の値を代入する．

1.2.4 体積効果のモデリング

一般に，現実のレオロジー物体は，物体変形による体積の変化が小さいという性質を持つ．しかし，三要素モデルによるトラス構造のみで構築した仮想物体は，外力を加えて変形させると体積が大きく変化する．そこで物体の体積変化に応じて内圧を発生するフォークトモデルを仮定し，自然状態の体積 V_{init} と現在の体積 V の値に応じて，図 1.6-(a),(b) に示すように物体表面部に力を発生させることで，体積効果を表現する．

体積の時間微分を \dot{V} とする．体積効果の弾性係数および粘性係数を，それぞれ k_{vol} および c_{vol} とする．ここで内圧 p を次式で与える．

$$p = -k_{vol}(V - V_{init}) - c_{vol}\dot{V} . \quad (1.13)$$

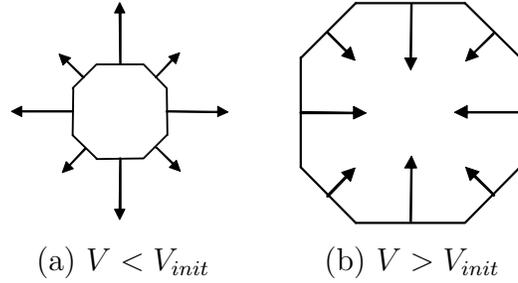


図 1.6: 体積効果

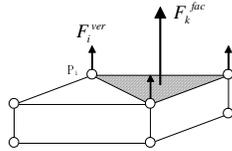


図 1.7: 体積効果によって発生する力

三角面の番号を k とし, その三角面の面積および物体の外側へ向かう単位法線ベクトルをそれぞれ S_k, \mathbf{n}_k^{out} とおく. 図 1.7 に示すように, 体積効果によって第 k 面に生じる力および質点 P_i に生じる力を, それぞれ $\mathbf{F}_k^{fac}, \mathbf{F}_i^{ver}$ とおく. このとき第 k 面に生じる力 \mathbf{F}_k^{fac} は,

$$\mathbf{F}_k^{fac} = pS_k\mathbf{n}_k^{out} \quad (1.14)$$

となる. 各面に生じる力は, 三角面を構成する三つの質点に均等に割り振られる. したがって, 質点の番号を i とし, i 番の質点 P_i を含む三角面の集合を A_i とすると, P_i に生じる力 \mathbf{F}_i^{ver} は, 次式で表される.

$$\mathbf{F}_i^{ver} = \sum_k \frac{\mathbf{F}_k^{fac}}{3}. \quad (1.15)$$

本節で提案した体積効果によって生じる力ベクトルの全質点における総和は 0 ベクトルであり, 運動量保存則や発生力が内力であるという条件を満たす.

体積効果のパラメータ k_{vol}, c_{vol} を一定以上の値に設定すると, 物体の変形中において物体の体積が常に初期体積のままに保たれる. このとき, 物体の変形特性は体積効果のパラメータに依存しない. つまり, 本節で述べた体積効果は, 常に一定の体積を保つ物体に対しては k_{vol} および c_{vol} のパラメータ同定を必要としない.

本節の議論は二次元物体に対しても同様に適用できる. 物体の体積および三角面の面積を, それぞれ二次元物体の面積および稜線の長さに置き換えることで, 二次元物体における面積効果を実現することが可能である.

1.2.5 同一モデルによる弾性，粘塑性，レオロジー物体の表現

本報告で示したモデルを用いることで，レオロジー物体を含む，弾性物体から粘塑性物体までをシームレスに表現することができる．粘塑性物体は，三要素モデルのパラメータであるフォークト部の弾性係数 k_1 を 0 とすることで表現できる．この時，三要素モデルは直列する二つのダンパとなっており，全体の粘性係数は $c_1 c_2 / (c_1 + c_2)$ で与えられる．弾性物体は，理論上，独立ダンパ部の粘性係数 c_2 を無限大とすることで表現できると考えられる．この時，三要素モデルは単独のフォークトモデルとなる．数値計算上は， c_2 を十分に高い値とすることで近似的に弾性物体を表現する．それに加え，フォークト比率制約条件 a_{min}, a_{max} を共に 1 とすることで，残留変位を生じることなく長時間のシミュレーションを行うことが可能である．

ここで， c_2 に極端に高い値を用いた場合の数値計算上の安定性について述べる．三要素モデルの二つのダンパの粘性係数がある一定の条件を超えると，系が発散する．これは，離散時間における計算を原因とするものである．ルンゲクッタ法におけるきざみ時間を T ，質点の質量を m ，安定条件に関する比例定数を A^{stable} とおくと，安定に計算を行うための条件は，

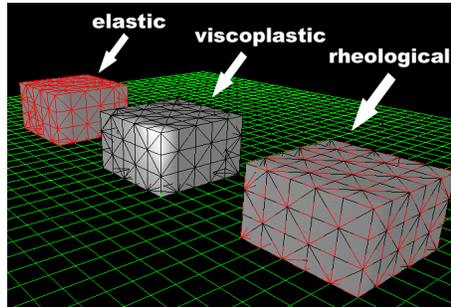
$$\frac{c_1 c_2}{c_1 + c_2} < A^{stable} \times \frac{m}{T} \quad (1.16)$$

となる．上式から， c_1 および c_2 の両方を同時に高い値に設定した場合，系が発散しやすくなることが分かる．それに対し， c_1 あるいは c_2 の一方のみを高い値に設定した場合，(1.16) 式の左辺では低い方の粘性係数の値が支配的となるため，系が発散しにくい．よって，本節における弾性物体の変形計算は安定である．

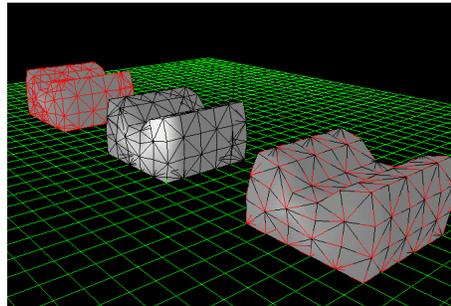
図 1.8 に，同一空間内に弾性物体，粘塑性物体，レオロジー物体を構築した例を示す．図 1.8-(a) が初期形状，図 1.8-(b) が一定の変位を与えたときの変形形状，図 1.8-(c) が変位を開放した後の定常状態における形状を表す．三種類の物体それぞれの特徴が表現されていることがわかる．

1.3 物体間の相互作用のモデリング

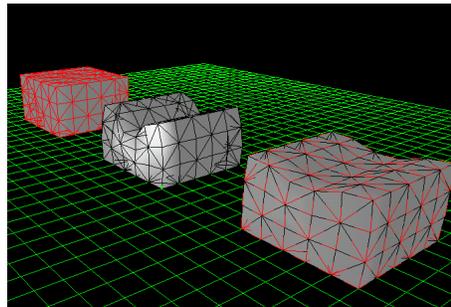
本節では，衝突と摩擦をモデリングする．衝突を表現する手法として，侵入量に比例した力を発生させるペナルティ法がある [10, 13]．しかし，ペナルティ法では物体間の干渉が解消されない場合がある．そこで本報告では，ペナルティ法に侵入量の時間積分によって発生する力を追加した手法を用いる．



(a) initial



(b) deformed



(c) stationary

図 1.8: 同一の仮想空間内への弾性物体，粘塑性物体，レオロジー物体の構築

1.3.1 従来のペナルティ法

ペナルティ法では干渉が生じた時に，干渉を解消する人工力を発生させる．図 1.9 に示すように，物体 A の頂点 R が物体 B の稜線 $Q_i Q_j$ を通ることにより，物体 A と B の干渉が生じているとする．稜線 $Q_i Q_j$ と頂点 R の距離を L_c ，距離方向の相対速度を V_c とする．従来のペナルティ法では， L_c の値を負とし，頂点 R に作用する人工力の大きさ F_c を，

$$F_c = -K_c L_c - C_c V_c \quad (1.17)$$

で与える．ここで， K_c は弾性係数， C_c は粘性係数を表す．従来のペナルティ法では，重力や電磁気力等の物体力により干渉が生じた場合，干渉は完全には解消されない．例えば，物体を床に置いた場合に，物体が床にめりこんだままとなる．これは，重力と釣り合うための人工力を発生させるために，侵入量 L_c が必要となるためである．

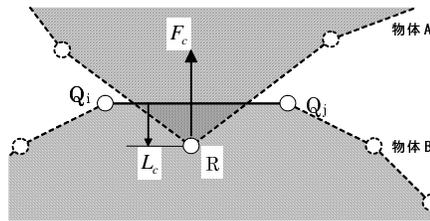


図 1.9: ペナルティ法

1.3.2 ペナルティ時間積分法

物体にめり込んだ頂点が，時間が経つにつれて稜線に近づくようにするために，ペナルティ法に侵入量の時間積分の項を加える．この手法をペナルティ時間積分法と呼ぶ．ペナルティ時間積分法での人工力 F_{co} は，

$$F_{co} = -K_c L_c - C_c V_c - I_c \int_{t_0}^{t_1} L dt \quad (1.18)$$

と表される．ここで， I_c は積分係数， t_0 は衝突判定領域に侵入した時間， t_1 は現在の時間である．

時間積分の値は，衝突判定領域から離れると，初期値 0 に戻るものとする．離散的な計算では，稜線上での人工力 F_{co} と物体力は必ずしも等しくならず， F_{co} が人工力より大きい場合，頂点は稜線の外側へ離れてしまう．そうなると，時間積分の値が初期値に戻ってしまうため，頂点 R は振動的な挙動を示す．そこで，図 1.10 に示すように，衝突判定領域を微小な長さ δ だけ稜線の外側に拡張し，その間に頂点が

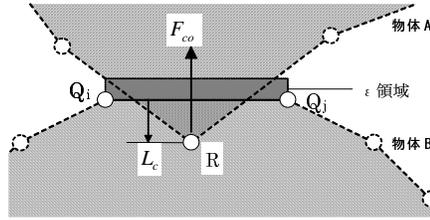


図 1.10: ペナルティ時間積分法

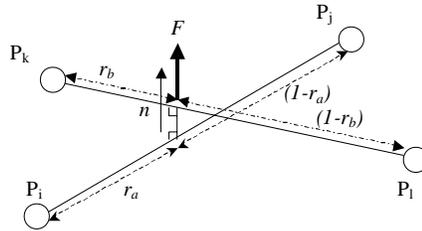


図 1.11: 三次元での稜線間の衝突における力の配分

ある場合も時間積分を続ける．その領域を ϵ 領域とする．これにより頂点 R は稜線近傍に収束する．物体の鋭角な部分に質点が衝突した場合における質点の貫通を減らすために，(1.18) 式の右辺における $-K_c L_c$ を $-K_c(L_c - \epsilon)$ に置き換える．これは進入量が低いほど高い弾性係数を持つ非線形な要素として説明される．ペナルティ時間積分法によって発生した力は，1.2.3 節で述べた手法に基づいて，質点に分配する．逆に 1.2.3 節で述べた位相保持アルゴリズムは，ペナルティ法に符号付距離の概念を導入したものであると考えることもできる．

三次元の場合は稜線と稜線の衝突も起こり得る．稜線と稜線の干渉の場合も同様に考えることができる．図 1.11 に示す稜線 $P_i P_j$ と稜線 $P_k P_l$ が衝突する場合，図 1.11 に示すようにそれぞれを内分する比率を $r_a : (1 - r_a)$ ， $r_b : (1 - r_b)$ とし，衝突によって発生する力の大きさを F ，両稜線に直交する単位ベクトルを n とすると，各質点にかかる力 F_i, F_j, F_k, F_l は，

$$F_i = -(1 - r_a)F n$$

$$F_j = -r_a F n$$

$$F_k = (1 - r_b)F n$$

$$F_l = r_b F n$$

となる．

1.3.3 動摩擦のモデリング

衝突により発生する力 F_{co} を垂直抗力とみなすことで，クーロン・アモントン則により，摩擦力を表現できる．動摩擦係数を μ とした場合，動摩擦力 F_{dr} は，

$$F_{dr} = \mu F_{co} \quad (1.19)$$

となる．質点 R に対し，相対速度を打ち消す方向に F_{dr} を与える．作用反作用の法則より，拮抗する力を分配して頂点 Q_i, Q_j に与える．これにより動摩擦力を表現できる．三次元の場合も同様に表すことができる．

1.4 シミュレーション

本章で提案した手法によって構築した仮想レオロジー物体の変形計算例を図 1.12-(a) ~ (d) に示す．これは，食品加工を想定し，コンベア上を流れるレオロジー物体を，ローラーによって成形する過程をシミュレートしたものである．ローラーは，1.2.5 節で提案した手法による硬い弾性物体を用いた．形状は正四角柱とし，コンベアの移動と同期して回転させた．図 1.12-(d) から，仮想レオロジー物体が実際のレオロジー物体と同様に，ローラーによって薄く伸ばされ，円形に近づいている様子が分かる．

次に，リアルタイムでの変形計算例として，マウスによる対話操作を行った仮想レオロジー物体変形シミュレーションの動作例を図 1.13-(a) ~ (c) に示す．計算は AthlonXP 2400+ 搭載 PC で行い，実時間で変形計算が可能な仮想物体の規模を測定した．三要素モデルによる稜線の計算のみを行った場合と，それに加えて，位相保持，体積効果，力依存型非線形ダンパ [24] および操作点との接触処理を追加して変形計算を行った場合について，100Hz もしくは 1000Hz の更新頻度で計算することができた物体の規模を表 1.1 に示す．ハプティックデバイスを用いて柔軟物の力覚提示を行うためには，数 10Hz から数 100Hz 程度の更新頻度による変形計算が必要である．本報告で提案したモデルでは，表に示した程度の小規模な物体であれば，力覚提示に必要な時間分解能での変形計算を行うことが可能であるということが分かった．

1.5 結言

本章では仮想レオロジー物体のモデリング手法を提案した．同一のモデルによる弾性から粘塑性までの統一的なモデリングを実現し，ペナルティ法に準じたアルゴリズムを用いることで物体間の衝突のモデリングとトラス構造モデルにおける位相

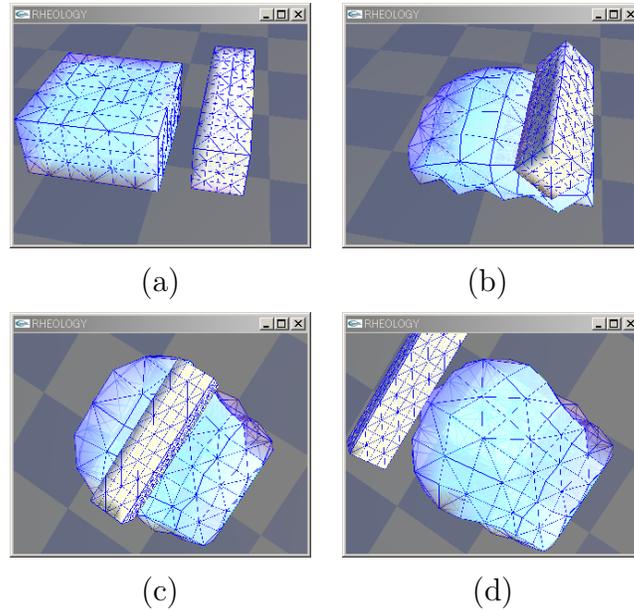


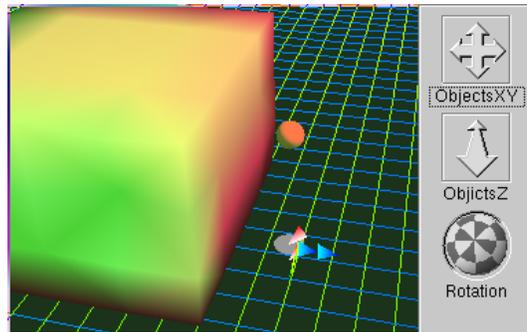
図 1.12: ローラーによる仮想レオロジー物体の成形

表 1.1: 実時間計算可能な物体規模

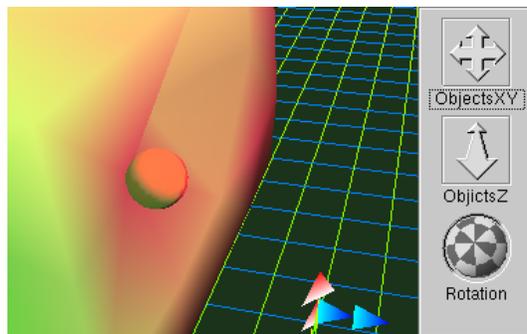
更新頻度 [Hz]	追加要素	各軸の質点数 (x,y,z)	総稜線数
100	ON	7,7,6	1385
100	OFF	9,8,8	2857
1000	ON	4,4,4	252
1000	OFF	6,5,5	661

保持を実現した．また時間積分項の導入によってペナルティ法の欠点である物体同士のめり込みを改善した．

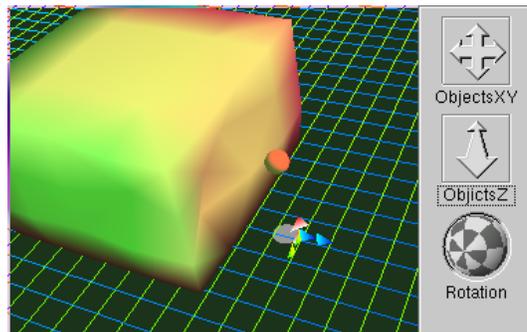
本章で提案したモデルを用いることで広い範囲の物体を表現することが可能である．ただし，実時間で計算することができる要素の数は十分ではなく，複雑な形状の物体を扱うことは困難である．しかし近年，FPGA (Field Programmable Gate Array) と呼ばれる書き換え可能 LSI が飛躍的な進歩を遂げている．本章で提案したモデルは並列処理に適しており，LSI を用いた並列パイプライン処理が，多数の要素を持つ複雑な仮想物体の実時間計算への有力な選択肢となっている．



(a) initial



(b) contact



(c) deformed

図 1.13: リアルタイム変形シミュレーション

実時間計算可能な仮想レオロジー物体の構築

友國 誠至^{*1} 杉山 勇太^{*1} 平井 慎一^{*2}

Construction of Virtual Rheological Objects Computable in Realtime

Seiji Tomokuni^{*1}, Yuuta Sugiyama^{*1}, and Shinichi Hirai^{*2}

Abstract – We describe virtual rheological objects computable in realtime. Deformable soft objects such as food and tissue show both elastic and plastic properties, which are referred to as rheological objects. Construction of virtual rheological objects has not been, however, established yet. In this article, we construct virtual rheological objects, which are applicable to realtime presentation of deformation and force of rheological objects. First, we formulate three-element model to describe the rheological deformation using a truss model. Second, applying an extended penalty method, we maintain the topology of the truss model and describe the interaction among objects. Next, we demonstrate coherent description of rheological objects, viscoelastic objects, and plastic objects. Finally, we demonstrate the realtime computation of virtual rheological objects.

Keywords : rheology, modeling, deformation, virtual object, contact

1 はじめに

これまで、バーチャルリアリティの分野では、粘弾性物体や塑性物体のモデリングに関する研究が多くなされてきた。しかし、現実世界には、粘弾性物体と塑性物体の中間的な性質を示すレオロジー物体が数多く存在する。レオロジー物体の例としては、パン生地やゼリー等の食品類、臓器や血管などの生体組織、粘土が挙げられる。これらを仮想空間に構築することで、手術、陶芸、食品加工などの幅広い分野におけるシミュレータへの応用が可能であると考えられる。しかしながら、レオロジー物体のモデリングに関する研究は、これまで成されていない。本論文では、仮想レオロジー物体を構築することを目的として、レオロジー物体のモデリングを行う。

1.1 レオロジー物体の性質

図 1-(a) に示す初期形状を有する物体に外力を作用させると、図 1-(b) に示すように変形すると仮定する。粘弾性物体では、図 1-(c) に示すように、外力を解放したときの形状が初期形状に一致する。外力を作用させた形状と外力解放後の形状との差を、戻り変位、初期形状と外力解放後の形状との差を残留変位とよぶ。粘弾性物体では、戻り変位があり、残留変位はない。塑性物体では、図 1-(d) に示すように、外力を解放したときの形状が変形形状に一致する。すなわち、塑性

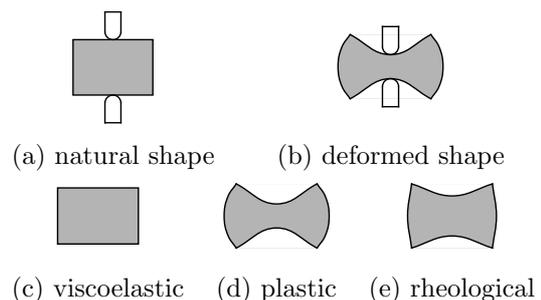


図 1 粘弾性物体，塑性物体，レオロジー物体
Fig.1 Viscoelastic, plastic, and rheological objects

物体では、残留変位があり、戻り変位はない。図 1-(e) に示すように、戻り変位と残留変位の両方を有する物体を、レオロジー物体とよぶ。

1.2 先行研究

現在、対話操作に適した実時間変形計算の可能な仮想弾性物体のモデリングに関する研究が精力的に行なわれている [1]-[6]。しかしこれらの研究で取り扱う物体は粘弾性物体に限られている。一方、材料学の分野においては古くからレオロジー物体の特性に関する研究がなされてきた [7]。しかし、これは全て一次元における材料特性を対象としている。プラスチック材料の分野では、分子レベルにおけるレオロジー物体の動力学シミュレータの開発が進められている [8]。しかし、これは形状を対象としておらず、対話操作に直接適用することはできない。レオロジー物体の変形モデルとしては、徳本らによるモデルが挙げられる [9]。しかし、このモデルでは変形形状が実際のレオロジー物

^{*1}立命館大学大学院 理工学研究科

^{*2}立命館大学 ロボティクス学科

^{*1}Graduate School of Science and Engineering, Ritsumeikan Univ.

^{*2}Department of Robotics, Ritsumeikan Univ.

体と大きく異なり、また、物体の大変形の際に変形計算が不安定となる。以上のように現状では、対話操作を想定した、実時間での仮想レオロジー物体の動的な変形計算手法は、まだ確立していない。

2 仮想レオロジー物体のモデリング手法

本節では、残留変位や戻り変位を持つ力学要素を用いることで、仮想レオロジー物体のモデリングを行う。さらに位相保持および体積効果を追加することで、実レオロジー物体に近い変形挙動を実現する。

2.1 三要素モデル

三要素モデルは、レオロジー的特性を有する力学モデルの中で最もシンプルなものである。このモデルは図2に示すように、戻り変位を生じるフォークトモデルと残留変位を生じる独立したダンパの直列によって構成される。本研究ではレオロジー的性質を表現するための一次元要素としてこの三要素モデルを用いる。

三要素モデルの長さおよび三要素モデルのフォークト部の長さをそれぞれ l , l_{voigt} とし、フォークト部の粘性係数および弾性係数を k_1, c_1 , 独立ダンパ部の粘性係数を c_2 とする。独立ダンパ部の長さ l_{damper} は、次式で表される。

$$l_{damper} = l - l_{voigt} . \quad (1)$$

フォークト部に作用する力 f および独立ダンパ部に作用する力 f_{damper} は、

$$f = -k_1(l_{voigt} - L) - c_1(\dot{l}_{voigt}) \quad (2)$$

$$\begin{aligned} f_{damper} &= -c_2(\dot{l}_{damper}) \\ &= -c_2(\dot{l} - \dot{l}_{voigt}) \end{aligned} \quad (3)$$

となる。ここで、フォークト部に作用する力および独立ダンパ部にかかる力は等しいことから、(2), (3) 式より、フォークト部の長さの変化に関する式が導かれる。

$$\dot{l}_{voigt} = \frac{-k_1(l_{voigt} - L) + c_2\dot{l}}{c_1 + c_2} . \quad (4)$$

変形計算中に三要素モデルのパラメータが変化しないものと仮定し、定数 $A = -k_1/(c_1 + c_2)$, 定数 $B = c_2/(c_1 + c_2)$ を導入すると、(4) 式は次式の形となる。

$$\dot{l}_{voigt} = A(l_{voigt} - L) + B\dot{l} . \quad (5)$$

質点 P_i を始点とする三要素モデルの集合を R_i , 質点 P_j を終点とする三要素モデルの集合を S_j で表す。稜線に向きを付け、第 k 稜線の始点から終点に向かう単位ベクトルを e_k と表す。このとき、集合 R_i に含まれる三要素モデル E_k が、質点 P_i に加える力は $f_k e_k$

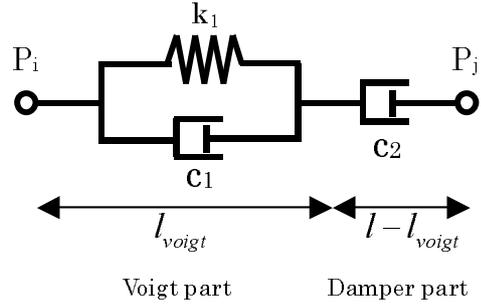


図2 三要素モデル
Fig.2 Three element model

に一致する。また、集合 S_j に含まれる三要素モデル E_k が、質点 P_j に加える力は $-f_k e_k$ に一致する。したがって、質点 P_i の運動方程式は、

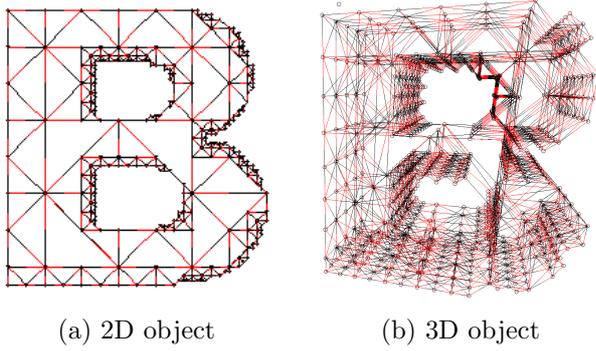
$$m_i \dot{v}_i = \sum_{k \in R_i} f_k e_k - \sum_{k \in S_i} f_k e_k + F_i^{ext} \quad (6)$$

と表される。ここで F_i^{ext} は、質点 P_i に作用する外力である。結局、物体モデルの運動方程式は、(5)(6) 式で与えられる。仮想レオロジー物体の変形は、運動方程式を数値的に解くことにより、計算することができる。本研究ではルンゲクッタ法を用いて計算を行う。

2.2 トラス構造

一次元要素である三要素モデルを格子状に組み合わせることによって、二次元物体および三次元物体を表現することが可能である。本研究ではグラフィック表示との相性に優れた三角面および四面体の集合によって、任意の形状を持つ二次元物体および三次元物体を構築する。仮想物体の変形計算に用いる要素として、質点、稜線、三角面および四面体を用意する。稜線、三角面、四面体は、向きを有する要素として扱う。二次元形状を現すデータとして三角面の集合、三角面を構成する各稜線と対向する頂点の組および稜線の向きを保持する。三次元形状を現すデータとして四面体の集合、四面体の各三角面と対向する頂点の組および三角面の向きを保持する。

図3に、トラス構造を用いて物体の形状を表現した例を示す。図3-(a)は2次元形状、図3-(b)は3次元形状の表現例である。図に示すように、大きさの異なる三角形あるいは四面体を用いて形状を表現する。図3の例では稜線上に質点が生成されているため、物体変形時にクラックと呼ばれる物体内部の亀裂を生じることがある。しかし、自然長を0としたフォークトモデルで質点と稜線を接続することで、クラックの発生を防止することができる。

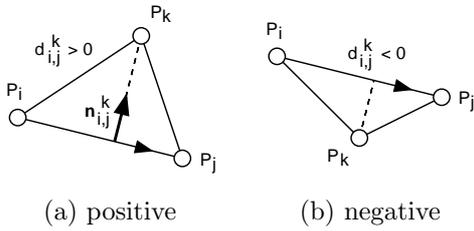


(a) 2D object

(b) 3D object

図3 トラス構造による形状表現の例

Fig. 3 Example of shape expression by truss model



(a) positive

(b) negative

図4 頂点と稜線の符号付き距離

Fig. 4 Signed distance between vertex and edge

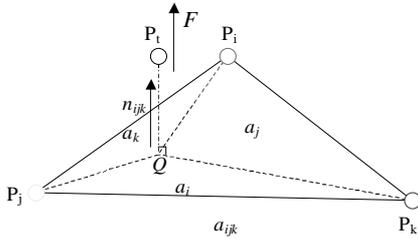


図5 三角面への力の配分

Fig. 5 Distribution of force to triangular face

2.3 位相保持

一次要素のトラス構造によるモデルでは、大変形の際に位相が崩れ、三角形や四面体などが反転することがある。その場合、元の形状に戻ろうとする復元力が失われ、三角形や四面体が鏡像の状態で安定するため、適切な戻り変位が得られない。そこで本節では、トラス構造モデルにおける位相的な接続関係を保つ手法を検討する。

二次元物体モデルにおいて位相的な接続関係が崩れるのは、図4(a),(b)に示すような鏡像どうしを区別できないことに起因する。そこで、鏡像どうしが区別できるように、個々の稜線に向きを付ける。稜線に向きが定義されているので、図4に示すように、三角要素の格子点 P_k と稜線 P_iP_j との間に、符号付距離を定義することができる。稜線 P_iP_j に垂直で、自然状態における P_k を向く単位ベクトルを n_{ij}^k 、稜線 P_iP_j

と格子点 P_k との符号付き距離を d_{ij}^k とする。位相的な接続関係を保つためには、符号付き距離 d_{ij}^k の値が小さくなったとき、質点 P_k に稜線から離れる方向の力を作用させる。本報告では、このような力が仮想的なフォークト要素により生成されると考え、以下に示す人工的な力を導入する。

$$f_{ij}^k = \begin{cases} 0 & (d_{ij}^k > \epsilon) \\ \{-K(d_{ij}^k - \epsilon) - Cd_{ij}^k\}n_{ij}^k & (d_{ij}^k \leq \epsilon) \end{cases} \quad (7)$$

ここで、 K, C はフォークト部の弾性係数、粘性係数、 ϵ は微小な正の定数である。符号付き距離 d_{ij}^k が閾値 ϵ より小さくなったとき、仮想的なフォークト要素が生成する力により、質点 P_k は稜線 P_iP_j より離れる方向に動く。

位相保持フォークト要素が発生する力 f_{ij}^k は質点 P_k に与えられる。作用反作用の法則により、稜線にも逆向きの力を与える。稜線は質量をもたないので、稜線にかかる力を稜線を構成する両端の質点 P_i と P_j に分配する必要がある。そこで、質点 P_k から稜線 P_iP_j へ下ろした垂線による稜線の内分点を Q とし、 Q 点によって P_iP_j を内分する比率 r_{ij}^k を定義する。そして質点 P_i および P_j に対して、それぞれ $r_{ij}^k f_{ij}^k, (1-r_{ij}^k) f_{ij}^k$ の力を与えることにより、 Q 点周りのモーメントが0となるように力が分配される。

しかし、 r_{ij}^k は負の値や、1を超える値になることがあり、三角面が極度に横に崩れた形状となった場合に、各質点に対して極端に大きな力が作用し、変形計算が不安定となることがある。そこで以下の拘束を加え、比率 r_{ij}^k の代わりに比率 R_{ij}^k を用いて力を分配する。

$$R_{ij}^k = \begin{cases} 0 & (r_{ij}^k < 0) \\ r_{ij}^k & (0 \leq r_{ij}^k \leq 1) \\ 1 & (r_{ij}^k > 1) \end{cases} \quad (8)$$

なお、三次元物体モデルにおいては、四面体要素の頂点と対向する三角形との間で、同様の符号付き距離が定義でき、上述の議論が適用できる。また反力を分配する方法も同様に三角形の面積比率を使い求めることができる。この時、二次元の場合と同様、四面体が極度に横方向に崩れた場合に変形計算が不安定とならないように面積比率に拘束を加える。図5に示すように、質点 P_i が三角形 $P_iP_jP_k$ と対向しているとする。質点 P_i の三角形 $P_iP_jP_k$ への足が三角形を内分する点を Q とする。三角形 $P_iP_jP_k$ の面積を a_{ijk} とし、三角形 QP_jP_k 、三角形 QP_kP_j 、三角形 QP_iP_j のそれぞれの符号付面積を a_i, a_j, a_k とする。各質点に対応する面積比率を r_i, r_j, r_k とする。まず、負の値を切り捨てた面積比率 r_i^{pos} を算出する。

$$r_i^{pos} = \begin{cases} 0 & (r_i \leq 0) \\ r_i & (r_i > 0) \end{cases} \quad (9)$$

同様に面積比率 r_j^{pos} , r_k^{pos} を求める．さらに，比率の合計が 1 となるよう，拘束後の面積比率 R_i を決定する．

$$R_i = \frac{r_i}{r_i^{pos} + r_j^{pos} + r_k^{pos}} \quad (10)$$

同様に R_j, R_k を求める．位相保持フォークト要素により，質点 P_i に作用する力を f_{ijk}^t とする．このとき，質点 P_i, P_j, P_k に，それぞれ $R_i f_{ijk}^t, R_j f_{ijk}^t, R_k f_{ijk}^t$ の力を分配する．

また位相的な接続関係が崩れる原因で，三要素モデル特有の問題として，フォークト部と独立ダンパ部の長さの比率がある．三要素モデルの長さを l とすると，フォークト部の長さ l_{voigt} は次式に示す条件を満たさなくてはならない．

$$0 \leq l_{voigt} \leq l \quad (11)$$

しかし， l_{voigt} は計算過程で (11) 式で示された範囲を超える可能性がある．そこで 0 から 1 までの値をとる定数 a_{min}, a_{max} をおき， l_{voigt} に以下の制約条件を課す．

$$la_{min} \leq l_{voigt} \leq la_{max} \quad (12)$$

計算過程で l_{voigt} の値が la_{min} を下回った時には l_{voigt} に la_{min} の値を代入し， l_{voigt} の値が la_{max} の値を上回った場合には， l_{voigt} に la_{max} の値を代入する．

2.4 体積効果のモデリング

一般に，現実のレオロジー物体は，物体変形による体積の変化が小さいという性質を持つ．しかし，三要素モデルによるトラス構造のみで構築した仮想物体は，外力を加えて変形させると体積が大きく変化する．そこで物体の体積変化に応じて内圧を発生するフォークトモデルを仮定し，自然状態の体積 V_{init} と現在の体積 V の値に応じて，図 6-(a),(b) に示すように物体表面部に力を発生させることで，体積効果を表現する．

体積の時間微分を \dot{V} とする．体積効果の弾性係数および粘性係数を，それぞれ k_{vol} および c_{vol} とする．ここで内圧 p を次式で与える．

$$p = -k_{vol}(V - V_{init}) + c_{vol}\dot{V} \quad (13)$$

三角面の番号を k とし，その三角面の面積および物体の外側へ向かう単位法線ベクトルをそれぞれ S_k, n_k^{out} とおく．図 7 に示すように，体積効果によって第 k 面に生じる力および質点 P_i に生じる力を，それぞれ F_k^{fac}, F_i^{ver} とおく．このとき第 k 面に生じる力 F_k^{fac} は，

$$F_k^{fac} = pS_k n_k^{out} \quad (14)$$

となる．各面に生じる力は，三角面を構成する三つの質点に均等に割り振られる．したがって，質点の番号

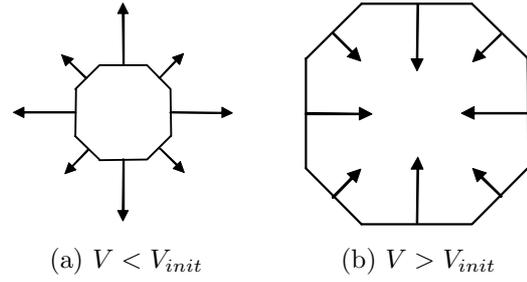


図 6 体積効果
Fig. 6 Volume effect

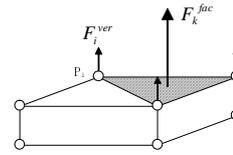


図 7 体積効果によって発生する力
Fig. 7 Force generated by volume effect

を i とし， i 番の質点 P_i を含む三角面の集合を A_i とすると， P_i に生じる力 F_i^{ver} は，次式で表される．

$$F_i^{ver} = \sum_{k \in A_i} \frac{F_k^{fac}}{3} \quad (15)$$

本節で提案した体積効果によって生じる力ベクトルの全質点における総和は 0 ベクトルであり，運動量保存則や発生力が内力であるという条件を満たす．

体積効果のパラメータ k_{vol}, c_{vol} を一定以上の値に設定すると，物体の変形中において物体の体積が常に初期体積のままに保たれる．このとき，物体の変形特性は体積効果のパラメータに依存しない．つまり，本節で述べた体積効果は，常に一定の体積を保つ物体に対しては k_{vol} および c_{vol} のパラメータ同定を必要としない．

本節の議論は二次元物体に対しても同様に適用できる．物体の体積および三角面の面積を，それぞれ二次元物体の面積および稜線の長さに置き換えることで，二次元物体における面積効果を実現することが可能である．

2.5 同一モデルによる粘弾性，粘塑性，レオロジー物体の表現

本論文で示したモデルを用いることで，レオロジーを含む，粘弾性物体から粘塑性物体までをシームレスに表現することができる．

粘塑性物体は，三要素モデルのパラメータであるフォークト部の弾性係数 k_1 を 0 とすることで表現できる．この時，三要素モデルは直列する二つのダンパとなっており，全体の粘性係数は $c_1 c_2 / (c_1 + c_2)$ で与えられる．

粘弾性物体は，理論上，独立ダンパ部の粘性係数 c_2

を無限大とすることで表現できると考えられる．この時，三要素モデルは単独のフォークトモデルとなっている．数値計算上は， c_2 を十分に高い値とすることで近似的に粘弾性物体を表現する．それに加え，フォークト比率制約条件 a_{min}, a_{max} を共に 1 とすることで，残留変位を生じることなく長時間のシミュレーションを行うことが可能である．

ここで， c_2 に極端に高い値を用いた場合の数値計算上の安定性について述べる．三要素モデルの二つのダンパの粘性係数がある一定の条件を超えると，系が発散する．これは，離散時間における計算が原因である．ルンゲクッタ法におけるきざみ時間を T ，質点の質量を m ，安定条件に関する比例定数を A^{stable} とおくと，安定に計算を行うための条件は

$$\frac{c_1 c_2}{c_1 + c_2} < A^{stable} \times \frac{m}{T} \quad (16)$$

となる．上式から， c_1 および c_2 の両方を同時に高い値に設定した場合，系が発散しやすくなることが分かる．それに対し， c_1 あるいは c_2 の一方のみを高い値に設定した場合，(16) 式の左辺では低い方の粘性係数の値が支配的となるため，系が発散しにくい．よって，本節における粘弾性物体の変形計算は安定である．

図 8 に，同一空間内に粘弾性物体，粘塑性物体，粘弾性，レオロジー物体を構築した例を示す．図 8-(a) が初期形状，図 8-(b) が一定の変位を与えたときの変形形状，図 8-(c) が変位を開放した後の定常状態における形状を表す．それぞれの特性が表現されていることがわかる．

3 物体間の相互作用のモデリング

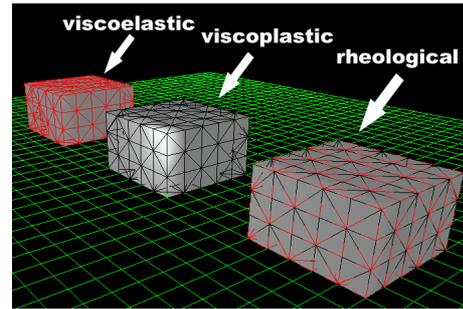
本節では，衝突と摩擦をモデリングする．衝突を表現する手法として，侵入量に比例した力を発生させるペナルティ法がある [10][11]．しかしペナルティ法では物体がめり込む場合がある．本論文では，ペナルティ法に侵入量の時間積分によって発生する力を追加した手法を用いる．

3.1 従来のペナルティ法

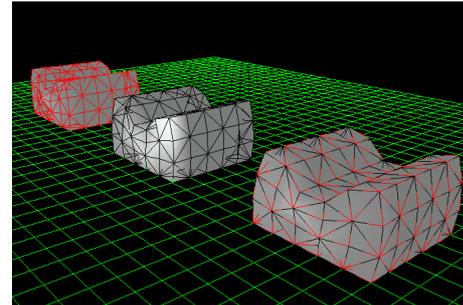
ペナルティ法では干渉が生じた時に，干渉を解消する人工力を発生させる．図 9 に示すように，物体 A の頂点 R が物体 B の稜線 $Q_i Q_j$ を通ることにより，物体 A と B の干渉が生じているとする．稜線 $Q_i Q_j$ と頂点 R の距離を L ，距離方向の相対速度を V とする．従来のペナルティ法では， L の値を負とし，頂点 R に作用する人工力の大きさ F_c を，

$$F_c = -K_c L - C_c V \quad (17)$$

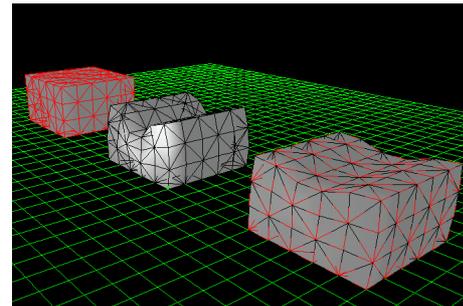
で与える．ここで， K_c は弾性係数， C_c は粘性係数を表す．従来のペナルティ法では，重力や電磁気力等の



(a) initial



(b) deformed



(c) stationary

図 8 同一の仮想空間内への粘弾性物体，粘塑性物体，レオロジー物体の構築

Fig. 8 Modeling of viscoelastic object, viscoplastic object, and rheological object in same virtual space

物体力により干渉が生じた場合，干渉は完全には解消されない．例えば，物体を床に置いた場合に，物体が床にめりこんだままとなる．これは，重力と釣り合うための人工力を発生させるために，侵入量 L が必要となるためである．

3.2 ペナルティ時間積分法

物体にめり込んだ頂点が，時間が経つにつれて稜線に近づくようにするために，ペナルティ法に侵入量の時間積分の項を加える．この手法をペナルティ時間積分法と呼ぶ．ペナルティ時間積分法での人工力 F_{co} は，

$$F_{co} = -K_c L - C_c V - I_c \int_{t_0}^{t_1} L dt \quad (18)$$

と表される．ここで， I_c は積分係数， t_0 は衝突判定領域に侵入した時間， t_1 は現在の時間である．

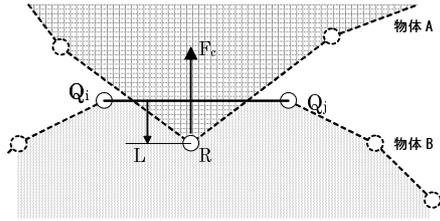


図9 ペナルティ法
Fig.9 penalty method

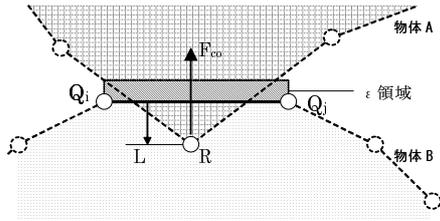


図10 ペナルティ時間積分法
Fig.10 Time-integrated penalty method

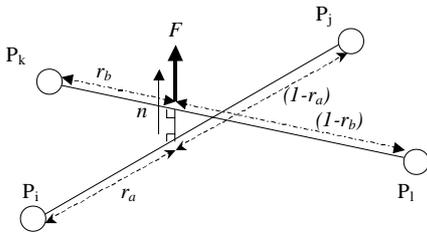


図11 三次元での稜線間の衝突における力の配分
Fig.11 Distribution of force in collision of edges

時間積分の値は、衝突判定領域から離れると、初期値0に戻るものとする。離散的な計算では、稜線上での人工力 F_{co} と物体力は必ずしも等しくならず、 F_{co} が人工力より大きい場合、頂点は稜線の外側へ離れてしまう。そうなると、時間積分の値が初期値に戻ってしまうため、頂点 R は振動的な挙動を示す。そこで、図10に示すように、衝突判定領域を微小な長さだけ稜線の外側に拡張し、その間に頂点がある場合も時間積分を続ける。その領域を ϵ 領域とする。これにより頂点 R は稜線近傍に収束する。

物体の鋭角な部分に質点が衝突した場合における質点の貫通を減らすために、(18)式の右辺における $-K_c L$ を $-K_c(L - \epsilon)$ に置き換える。これは進入量が低いほど高い弾性係数を持つ非線形な要素として説明される。

ペナルティ時間積分法によって発生した力は、2.3節で述べた手法に基づいて、質点に分配する。逆に2.3節で述べた位相保持アルゴリズムは、ペナルティ法に符号付距離の概念を導入したのものであると考えることもできる。

三次元の場合は稜線と稜線の衝突も起こり得る。稜線と稜線の干渉の場合も同様に考えることができる。図11に示す稜線 $P_i P_j$ と稜線 $P_k P_l$ が衝突する場合、図11に示すようにそれぞれを内分する比率を $r_a : (1 - r_a), r_b : (1 - r_b)$ とし、衝突によって発生する力の大きさを F 、両稜線に直交する単位ベクトルを n とすると、各質点にかかる力 F_i, F_j, F_k, F_l は、

$$\begin{aligned} F_i &= -(1 - r_a)F n \\ F_j &= -r_a F n \\ F_k &= (1 - r_b)F n \\ F_l &= r_b F n \end{aligned} \quad (19)$$

となる。

3.3 動摩擦のモデリング

衝突により発生する力 F_{co} を垂直抗力とみなすことにて、クーロン・アモントン則により、摩擦力を表現できる。動摩擦係数を μ とした場合、動摩擦力 F_{dr} は

$$F_{dr} = \mu F_{co} \quad (20)$$

となる。質点 R に対し、相対速度を打ち消す方向に F_{dr} を与える。作用反作用の法則より、拮抗する力を分配して頂点 Q_i, Q_j に与える。これにより動摩擦力を表現できる。三次元の場合も同様に表すことができる。

4 シミュレーション

本論文で提案した手法によって構築した仮想レオロジー物体の変形計算例を図12-(a)~(d)に示す。これは、食品加工を想定し、コンベア上を流れるレオロジー物体を、ローラーによって成形する過程をシミュレートしたものである。ローラーは、2.5節で提案した手法による硬い粘弾性物体を用いた。形状は正四角柱とし、コンベアの移動と同期して回転させた。図12-(d)から、仮想レオロジー物体が実際のレオロジー物体と同様に、ローラーによって薄く伸ばされ、円形に近づいている様子が分かる。

次に、リアルタイムでの変形計算例として、マウスによる対話操作を行った仮想レオロジー物体変形シミュレーションの動作例を図13-(a)~(c)に示す。計算はAthlonXP 2400+搭載PCで行い、実時間で変形計算が可能な仮想物体の規模を測定した。三要素モデルによる稜線の計算のみを行った場合と、それに加えて、位相保持、体積効果、力依存型非線形ダンパ [9] および操作点との接触処理を追加して変形計算を行った場合について、100Hz もしくは 1000Hz の更新頻度で計算することができた物体の規模を表1に示す。ハプティックデバイスを用いて柔軟物の力覚提示を行うためには、数10Hz から数100Hz 程度の更新頻度によ

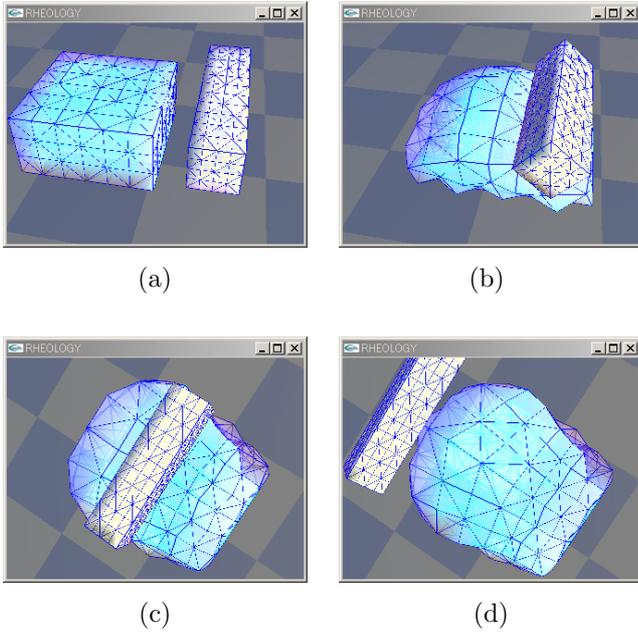


図 12 ローラーによる仮想レオロジー物体の成形
Fig. 12 Forming of virtual rheological object by roller

る変形計算が必要である．本論文で提案したモデルでは，表に示した程度の小規模な物体であれば，力覚提示に必要な時間分解能での変形計算を行うことが可能であるということが分かった．

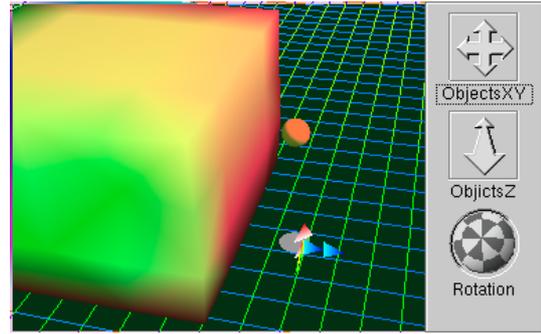
表 1 実時間計算可能な物体規模
Table 1 Object size computable in realtime

更新頻度 [Hz]	追加要素	各軸の質点数 (x,y,z)	総稜線数
100	ON	7,7,6	1385
100	OFF	9,8,8	2857
1000	ON	4,4,4	252
1000	OFF	6,5,5	661

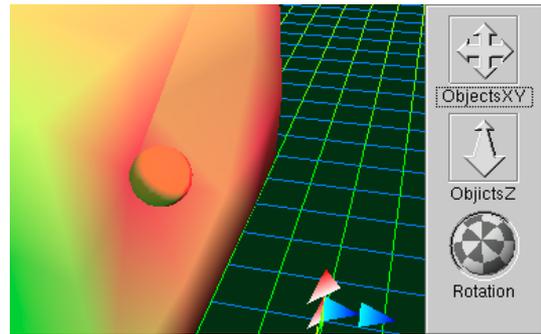
5 おわりに

本論文では仮想レオロジー物体のモデリング手法を提案した．同一のモデルによる粘弾性から粘塑性までの統一的なモデリングを実現し，ペナルティ法に準じたアルゴリズムを用いることで物体間の衝突のモデリングとトラス構造モデルにおける位相保持を実現した．また時間積分項の導入によってペナルティ法の欠点である物体同士のめり込みを改善した．

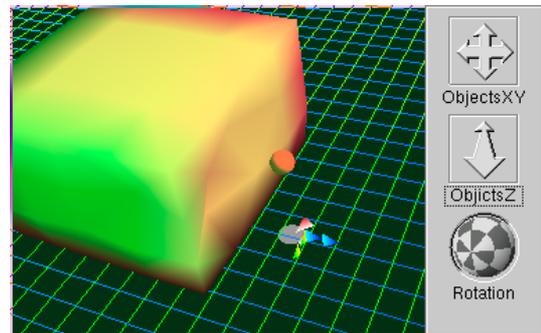
本論文で提案したモデルを用いることで広い範囲の物体を表現することが可能である．ただし，実時間で計算することができる要素の数は十分ではなく，複雑な形状の物体を扱うことは困難である．しかし近年，FPGA(Field Programmable Gate Array) と呼ばれる書き換え可能 LSI が飛躍的な進歩を遂げている．本論文で提案したモデルは並列処理に適しており，LSI を



(a) initial



(b) contact



(c) deformed

図 13 リアルタイム変形シミュレーション
Fig. 13 Realtime simulation of rheological deformation

用いた並列パイプライン処理が，多数の要素を持つ複雑な仮想物体の実時間計算への有力な選択肢となっている．

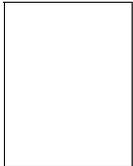
参考文献

- [1] D. Terzopoulos, J. Platt, A. Barr, K. Fleischer: Elastically Deformable Models, Computer Graphics, Vol.21, No4, pp205-214(1987)
- [2] 広田光一, 金子豊久: 仮想物体の弾性モデルに関する検討, 計測自動制御学会論文集, Vol.32, No.3, pp.232-238(1998)
- [3] 宮崎慎也, 吉田俊介, 安田孝美, 横井茂樹: 局所形状保持に基づく仮想弾性物体モデルの提案, 電子情報通信学会論文誌 A, Vol.J82-A, No.7, pp.1148-1155(1999)
- [4] S. Cotin, H. Delingette, N. Ayache: A Hybrid

- Elastic Model allowing Real-Time Cutting, Deformations and Force-Feedback for Surgery Training and Simulation, Visual Computer, Vol.16, No.8, pp.437-452(2000)
- [5] G. Picinbono, H. Delingette, N. Ayache: Non-linear and anisotropic elastic soft tissue models for medical simulation, Proc. 2001 IEEE Int. Conf. on Robotics and Automation Seoul, pp.1371-1376(2001)
- [6] Y. Zhuang, J. Canny: Haptic Interaction with Global Deformations, Proc. 2000 IEEE Int. Conf. on Robotics and Automation San Francisco, pp.2428-2433(2000)
- [7] 種谷真一著: やさしいレオロジー工学, 工業調査会(1990)
- [8] 増淵雄一: 分子レオロジーシミュレータ「NAPLES」, <http://masubuchi.jp/NAPLESWEB/>
- [9] 徳本真一, 平井慎一: 形状制御のためのレオロジー物体の変形モデリング, 日本ロボット学会誌, Vol.18, No.8, pp.87-94(2001)
- [10] H. Keller, H. Stolz, A. Ziegler, T. Braunl: Virtual Mechanics Simulation and Animation of Rigid Body System, <http://citeseer.nj.nec.com/keller94virtual.html>, p28(1993)
- [11] M. Moore, J. Wilhelms: Collision Detection and Response for Computer Animation, Computer Graphics, Vol.22, No.4, pp.289-298(1988)

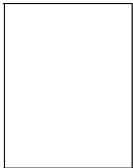
[著者紹介]

友國 誠至



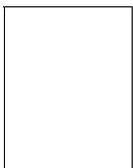
2003年立命館大学工学部卒業。現在、同大学大学院博士前期課程に在籍。仮想レオロジー物体のモデリングおよび高速計算に関する研究に従事。学士(工学)

杉山 勇太



2003年立命館大学工学部卒業。現在、同大学大学院博士前期課程に在籍。学士(工学)

平井 慎一



1990年 京都大学大学院工学研究科 博士課程数理工学専攻単位取得退学。同年 大阪大学工学部 電子制御機械工学科助手。1995年同助教授。1996年 立命館大学工学部 ロボティクス学科助教授, 2002年同教授となり, 現在に至る。柔軟物モデリング, リアルタイムビジョン, 分散マニピュレーションなどの研究に従事。博士(工学)

Constructing Rheologically Deformable Virtual Objects

Masafumi KIMURA, Yuuta SUGIYAMA, Seiji TOMOKUNI, and Shinichi HIRAI

Dept. of Robotics, Ritsumeikan Univ., Kusatsu, Shiga 525-8577, Japan

E-mail: hirai@se.ritsumei.ac.jp

http://www.ritsumei.ac.jp/se/~hirai/

Abstract

A physical modeling of rheological objects is presented. Objects showing rheological nature involve foods and biological tissues yet no systematic approach to build their virtual objects is not established. In this article, we will construct 2D/3D virtual rheological objects.

Keywords: *deformation, modeling, rheological objects*

1 Introduction

Deformable objects can be categorized into 1) viscoelastic objects, 2) plastic objects, and 3) rheological objects. Construction of virtual viscoelastic objects and virtual plastic objects has been studied extensively in computer-aided surgery and computer graphics. On the other hand, construction of virtual rheological objects has not been studied yet though various objects such as type of food and biological tissues in the real world tend to deform rheologically.

In this article, we will develop a systematic and coherent method to construct virtual rheological objects. First, we will summarize the properties of rheological deformation. Secondly, we will select rheological elements appropriate for the construction of virtual rheological objects. Thirdly, we will apply the particle-based modeling to virtual rheological objects. We will then investigate the topology maintenance of virtual rheological objects. Finally, we will simulate the physical interaction among rheological objects.

Related works Solid mechanics has also been studied for a long time to formulate the deformation of solid bodies [1]. Solid mechanics basically focuses on the local deformation of solid bodies rather than the global deformation of objects. Rheology has been studied for past several decades and fruitful results have been obtained [2]. Rheology focuses on one-dimensional deformation rather than two-dimensional (2D) or three-dimensional (3D) deformation as well.

Modeling of global object deformation has been extensively studied in computer graphics and virtual reality. Elasticity theory has been applied to the modeling of deformable objects in physically based modeling [3, 4]. Introduction of finite element method (FEM) has extended these works; geometrically-nonlinear FEM and rotation-invariant nonlinear FEM are applied to the modeling of global deformation [5, 6]. Explicit FEM approach with Green strain is proposed to

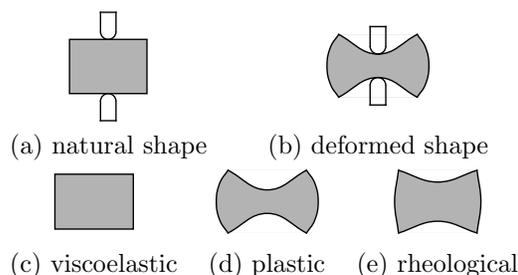


Figure 1: Viscoelastic, plastic, and rheological deformations

perform realtime computation of global deformation [7]. Boundary element method (BEM) has been introduced to the modeling of deformable objects [8, 9]. BEM approach is applicable to uniform objects alone but can reduce the computation time, resulting in real-time simulation of global object deformation. Non-linear shell theory has been applied to the modeling of fabric deformation [10]. Particle-based model of a cloth has been proposed for the drape simulation of the cloth [11]. Implicit numerical integration has been introduced to the particle-based cloth model to reduce the computation time [12]. Modeling of plastic objects has been studied in computer graphics [13] and has been applied to computer crafts [14, 15].

This article contributes to the modeling of rheologically deformable objects. We will first formulate the dynamic behavior of rheological elements. Particle-based modeling [16, 17] will be applied to describe the 2D/3D object deformation. Then, topology maintenance of virtual rheological objects will be addressed. Finally, penalty method [18, 19] will be applied to simulate the physical interaction among rheological objects.

2 Rheological Deformation

Objects deform in response to forces applied to the objects. Objects can be categorized into three groups with respect to their deformation. Assume that a natural shape of an object is as given in Figure 1-(a). On applying external forces, the object deforms as in Figure 1-(b). Let us release the applied force and exam-

Table 1: Inferring rules of residual deformation

serial	residual	non-residual
residual	residual	residual
non-residual	residual	non-residual

(a) serial

parallel	residual	non-residual
residual	residual	non-residual
non-residual	non-residual	non-residual

(b) parallel

Table 2: Inferring rules of bouncing deformation

serial	bouncing	non-bouncing
bouncing	bouncing	bouncing
non-bouncing	bouncing	non-bouncing

(a) serial

parallel	bouncing	non-bouncing
bouncing	bouncing	bouncing
non-bouncing	bouncing	non-bouncing

(b) parallel

ine the stable shape after the release. Deformation of *viscoelastic objects* is completely lost and their stable shape coincides with their natural shape, as illustrated in Figure 1-(c). Namely, viscoelastic objects have no *residual deformation*. Deformation of *plastic objects* completely remains and their stable shape coincides with their deformed shape under the applied forces, as shown in Figure 1-(d). Namely, plastic objects have no *bouncing deformation*. Objects with residual deformation and bouncing deformation are referred to as *rheological objects*. Deformation of rheological objects is partially lost after the applied forces are released, as illustrated in Figure 1-(e). Various objects including foods and tissues are categorized into rheological objects.

3 Modeling of Rheological Elements

3.1 Selection of Rheological Elements

Rheological objects deform according to forces applied to the objects. The relationship between the applied forces and the object deformation can be described in a physical model. Let us introduce an elastic element and a viscous element so that a physical model can describe the time-dependent deformation of a rheological object. Various deformation properties are then described by combinations of the two fundamental elements. These combinations are referred to as *rheological elements*. Next, we have to select rheological elements appropriate for virtual rheological objects.

As for the deformation properties of a rheological object, recall that 1) residual deformation is involved,

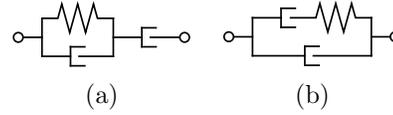


Figure 2: Minimal elements describing rheological deformation

2) bouncing displacement is involved, and 3) vibrations decrease. Let us examine whether individual rheological elements satisfy the first condition. Let us investigate a rheological element consisting of two elements connected in series. If either of the elements has residual deformation, the connected element has residual deformation as well. If neither of the two has residual deformation, the connected element has no residual deformation. These inferences are summarized in Table 1-(a). Let us investigate a rheological element consisting of two elements connected in parallel. If both of the elements have residual deformation, the connected element has residual deformation. If either of the elements has no residual deformation, the connected element has no residual deformation. These inferences are summarized in Table 1-(b). Note that a viscous element has residual deformation while an elastic element has no residual deformation. Thus, we can determine whether a given rheological element has residual deformation or not using Table 1.

Similarly, let us examine whether individual rheological elements satisfy the second condition. Let us investigate a rheological element consisting of two elements connected in series. If either of the elements has bouncing deformation, the connected element also has bouncing deformation. If neither of the two has bouncing deformation, the connected element has no bouncing deformation. These inferences are summarized in Table 2-(a). Let us investigate a rheological element consisting of two elements connected in parallel. If either of the elements has bouncing deformation, the connected element has bouncing deformation as well. If neither of the two elements has bouncing deformation, the connected element has no bouncing deformation. These inferences are summarized in Table 2-(b). Note that an elastic element has bouncing deformation while a viscous element has no bouncing deformation. Thus, we can determine whether a given rheological element has bouncing deformation or not using Table 2.

Now let us examine whether individual rheological elements satisfy the third condition. Note that elements connected in parallel have the same displacement. Thus, a set of elements connected in parallel is referred to as a *part* of a rheological element. Each rheological element can be then regarded as a series of parts. If a part involves viscous elements, any vibration on the part converges to zero. On the other hand, vibration on a part without viscous elements oscillates and does not converge to zero. Thus, we find that all parts must involve viscous elements to satisfy the third condition.

Consequently, we find that rheological elements listed in Figure 2 satisfy the three conditions and consist of the minimal fundamental elements. In this arti-

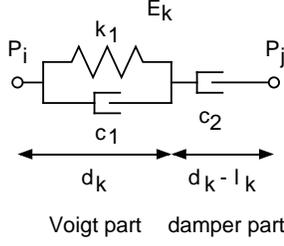


Figure 3: Three-element model

cle, we will use the rheological element shown in Figure 2-(a), which is referred to as a *three-element model*.

3.2 Formulation of the Rheological Element

We will apply the particle-based modeling [16, 17] to describe the deformation of rheological objects. We will describe a 2D/3D object shape by a combination of triangles/tetrahedra. Vertices of individual triangles/tetrahedra correspond to mass particles and their edges correspond to rheological elements. Then, the object model consists of a set of mass particles and a set of rheological elements among the particles. Let P_0 through P_{N-1} be mass particles, where N denotes the number of the particles. Let E_0 through E_{M-1} be rheological elements, where M specifies the number of the elements. Each rheological element connects two mass particles. One particle is referred to as the starting particle of the element while the other is referred to as its end particle.

Let us formulate the dynamic equation of a virtual rheological object. We will apply the three-element model, illustrated in Figure 2-(a), to rheological elements. Two mass particles P_i and P_j are connected by a rheological element E_k , as shown in Figure 3. This rheological element consists of two parts; the left part is referred to as a Voigt part and the right part is referred to as a damper part. Let P_i be the starting particle of the element while P_j be its end particle. Let \mathbf{x}_i be the position of particle P_i , \mathbf{v}_i be its velocity, and m_i be its mass. Let l_k be the length of element E_k and d_k be the length of its Voigt part. Then, the length of its damper part coincides with $l_k - d_k$. We find that state variables of the mechanical system shown in the figure are given by $\mathbf{x}_i, \mathbf{v}_i, \mathbf{x}_j, \mathbf{v}_j$, and d_k . Let L_k be the natural length of the Voigt part of element E_k . Since the extension of the Voigt part is given by $d_k - L_k$, the magnitude of a force generated by the Voigt part is described as follows:

$$f_k^{voigt} = -k_1\{d_k - L_k\} - c_1\dot{d}_k, \quad (1)$$

where k_1 represents the stiffness of the Voigt part and c_1 denotes its viscosity. Recalling that the length of the damper part is given by $l_k - d_k$, the magnitude of a force generated by the damper part is described as follows:

$$f_k = -c_2\{\dot{l}_k - \dot{d}_k\}, \quad (2)$$

where c_2 represents the viscosity of the damper part. Since the forces generated by the two parts coincide

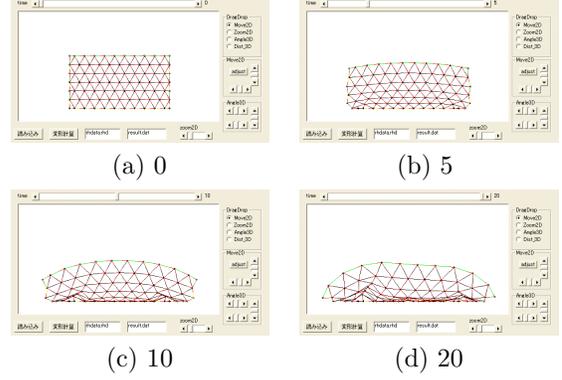


Figure 4: Simulation with linear damper

with each other, say, $f_k^{voigt} = f_k$, we have the following equation:

$$\dot{d}_k = \frac{-k_1\{d_k - L_k\} + c_2\dot{l}_k}{(c_1 + c_2)}. \quad (3)$$

Length l_k satisfies $l_k^2 = (\mathbf{x}_i - \mathbf{x}_j) \cdot (\mathbf{x}_i - \mathbf{x}_j)$. Differentiating this equation with respect to time yields

$$\dot{l}_k = \frac{(\mathbf{x}_i - \mathbf{x}_j) \cdot (\mathbf{v}_i - \mathbf{v}_j)}{l_k}. \quad (4)$$

Note that a force applied to particle P_i by element E_k is described as $f_k \mathbf{e}_k$ while a force applied to particle P_j is given by $-f_k \mathbf{e}_k$, where \mathbf{e}_k is a unit vector from the starting particle to the end particle, which is given by $\mathbf{e}_k = (\mathbf{x}_j - \mathbf{x}_i)/l_k$.

Let R_i be a set of rheological elements with particle P_i as their starting particle and S_i be a set of rheological elements with particle P_i as their end particle. Then, any element involved in set R_i applies force $f_k \mathbf{e}_k$ to mass particle P_i while any element involved in set S_i applies force $-f_k \mathbf{e}_k$ to the mass particle. Consequently, we find that the dynamic equation of particle P_i is described as follows:

$$m_i \dot{\mathbf{v}}_i = \sum_{k \in R_i} f_k \mathbf{e}_k - \sum_{k \in S_i} f_k \mathbf{e}_k + \mathbf{F}_i^{ext}, \quad (5)$$

where \mathbf{F}_i^{ext} be the resultant of external forces applied to particle P_i . On solving differential equations (3) and (5) numerically, we can compute the deformation of a rheological object.

3.3 Force-dependent Nonlinear Damper

Rheological elements involve damper parts to describe residual deformation. Conventional three-element model applies a linear damper to its damper part. Displacement of the linear damper continues increasing or decreasing as long as a force is exerted on the damper. This property is, however, inadequate for the introduction of gravity into virtual rheological objects. A virtual rheological object involving linear damper parts continues deforming as long as gravity forces are exerted on its mass particles. Thus, it

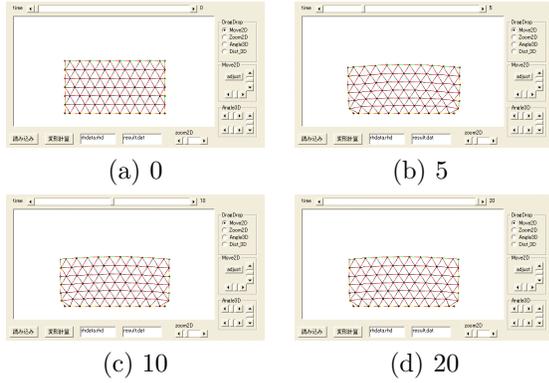


Figure 5: Simulation with force-dependent nonlinear damper

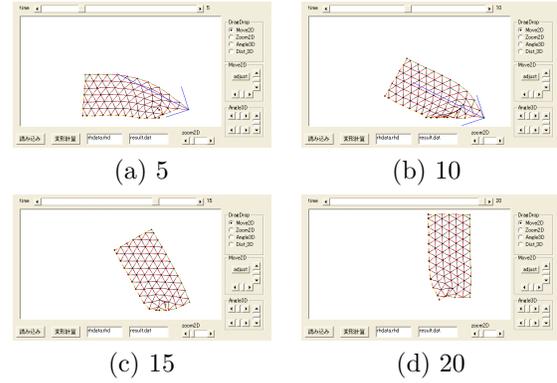


Figure 6: Simulation without gravity

turns out that the shape of a virtual rheological object collapses under gravity. Figure 4 simulates the deformation of a virtual rheological object involving linear damper parts under gravity. As shown in the figure, the object continues deforming and the shape of the object collapses finally. Thus, we will introduce a force-dependent nonlinear damper into the damper part of a three-element model. The viscosity of a force-dependent nonlinear damper changes according to the magnitude of a force applied to it. The viscosity corresponding to the gravity force applied to a particle must take a large value so that the shape of a virtual rheological object does not collapse under gravity. In this article, viscosity of a force-dependent nonlinear damper is given as follows:

$$c_2 = \begin{cases} c_{max} & (f \leq f_1) \\ Ae^{-Bf} & (f_1 \leq f \leq f_2) \\ c_{min} & (f \geq f_2) \end{cases},$$

where f denotes a force generated by the damper part, f_1 , f_2 , A , and B are constants, $c_{max} = Ae^{-Bf_1}$, and $c_{min} = Ae^{-Bf_2}$. Figure 5 simulates the deformation of a virtual rheological object involving force-dependent nonlinear dampers. We find that the shape of the object does not collapse under gravity.

On introducing force-dependent nonlinear dampers into a virtual rheological object, we can appropriately compute the deformation of the object regardless of body forces such as gravity and electromagnetic forces. Figures 6 and 7 show the deformation and the motion of a rheological object on a rigid table caused by an external force. Gravity does not work in Figure 6 while it works in Figure 7. Without gravity, only a reaction force from the table is applied to the object after the exerted external force is lost. Thus, the object moves upward, as shown in Figure 6-(d). Under gravity, the left bottom region of the object makes contact with the table again after the external force is lost because a reaction force and a gravitational force are applied to the object.

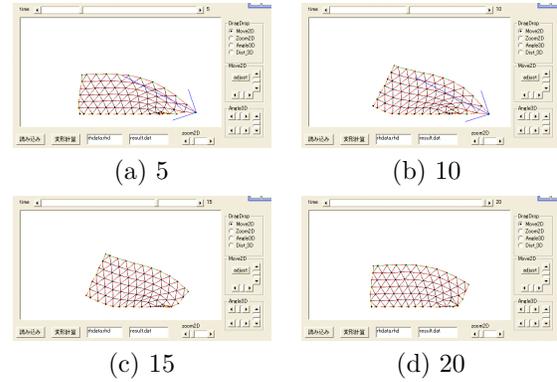


Figure 7: Simulation with gravity

4 Particle-based Modeling of Rheological Objects

4.1 Particle-based Model

We have introduced particle-based approach [16, 17] to describe 2D/3D deformation of rheological objects. In this section, we will describe the topology of a virtual rheological object. The particle-based model involves a set of mass particles and a set of rheological elements among the particles. Each element has its starting and end particles. This implies that each element is directed from the starting particle to the end particle. Consequently, topology of a virtual rheological object is described by a directed graph. Nodes in the graph represent mass particles and the arcs describe directed rheological elements. Moreover, a 2D/3D shape is described by a connection of triangles/tetrahedra, which must be involved in object topology. Figure 8 shows a simple example of a 2D discrete element model. This model consists of 4 mass particles, 5 rheological elements, and 2 triangles. Table 3 provides the topology of this model. The starting and end particles are specified for each element. Each triangle consists of three arcs in the positive direction or in the negative direction. The symbol following each element denotes its direction in the triangle.

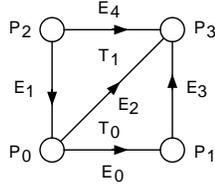


Figure 8: Example of 2D discrete element model

Table 3: Description of discrete element model

element	start point	end point
E_0	P_0	P_1
E_1	P_2	P_0
E_2	P_0	P_3
E_3	P_1	P_3
E_4	P_2	P_3

triangle	arcs					
T_0	E_0	+	E_3	-	E_2	-
T_1	E_2	+	E_4	-	E_1	+

4.2 Topology Maintenance

Topological connection among mass particles in a virtual rheological object must be consistent so that the deformation of the object can be computed appropriately. Since the dynamic equations given in eqs.(3) and (5) are solved numerically at discrete time points, the topological connection often collapses, resulting in failure of the computation.

Recall that a 2D virtual rheological object consists of triangles. In order to maintain the consistency in the topological connection, we have to distinguish a triangle from its reflection. Noting that edges of a triangle are directed, we find that a signed distance between a vertex of a triangle and its opposite edge can be defined, as illustrated in Figure 9. Let \mathbf{n}_{ij}^k be a unit vector perpendicular to edge P_iP_j and directed to vertex P_k at the natural state of a virtual rheological object, as shown in Figure 9-(a). The signed distance between vertex P_k and edge P_iP_j is then formulated as follows:

$$d_{ij}^k = \overrightarrow{P_iP_k} \cdot \mathbf{n}_{ij}^k. \quad (6)$$

This signed distance must be larger than a small positive value so that the topological connection among particles P_i , P_j , and P_k is consistent. In other words, when the signed distance is shorter than the small positive value, particle P_k must be guided so that the signed distance increases. Thus, we will introduce the following artificial force generated by a virtual Voigt model between vertex P_k and edge P_iP_j :

$$\mathbf{f}_{ij}^k = \begin{cases} \mathbf{0} & (d_{ij}^k > \epsilon) \\ \{-K(d_{ij}^k - \epsilon) - Cd_{ij}^k\} \mathbf{n}_{ij}^k & (d_{ij}^k \leq \epsilon) \end{cases}, \quad (7)$$

where K and C denote elasticity and viscosity of the

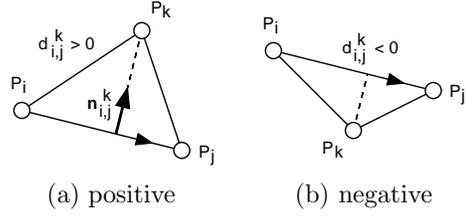


Figure 9: Signed distance between vertex and edge

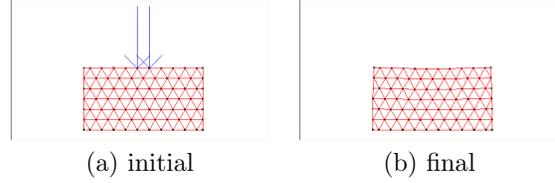


Figure 10: Modeling of viscoelastic object

virtual Voigt model, respectively, and ϵ represents a small positive threshold. When the signed distance d_{ij}^k is below threshold ϵ , the above artificial force is applied to particle P_k to increase the signed distance. The discussion can be extended to a 3D virtual rheological object by introducing a signed distance between a vertex of a tetrahedron and its opposite face.

The topological connection of a virtual rheological object may also collapse due to an inappropriate value of $a_k = d_k/l_k$, which specifies the ratio between the length of the Voigt part and that of the damper part. Note that ratio a_k must satisfy the condition $0 \leq a_k \leq 1$. Solving dynamic equations numerically at discrete time points often breaks this condition and the consistency in the topological connection is lost. Thus, we will define the possible minimum value a_{min} and the possible maximum value a_{max} of variable a_k . Namely, the following condition is imposed on ratio a_k :

$$a_{min} \leq a_k \leq a_{max}. \quad (8)$$

When the value of length d_k is below $a_{min}l_k$ during the computation process, the minimum value $a_{min}l_k$ is substituted into length d_k . When the value of length d_k exceeds $a_{max}l_k$ during the computation process, the maximum value $a_{max}l_k$ is substituted into length d_k .

5 Simulating Rheological Objects

5.1 Elasticity and plasticity

Our approach can describe not only rheological objects but viscoelastic objects and plastic objects in a systematic and coherent manner. Viscoelastic objects can be described by the Voigt part alone. Recall that the ratio of the Voigt part is described by a_k and the condition given in eq.(8) is imposed on the ratio. Thus, viscoelastic objects can be described simply by substituting 1 into both a_{min} and a_{max} . Figure 10 shows an example of the modeling of a viscoelastic object.

Plastic objects can be described by single dampers. On substituting 0 into elasticity k_1 of the Voigt part, a

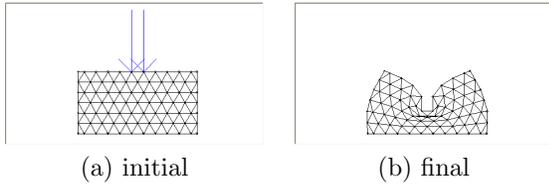


Figure 11: Modeling of plastic object

three-element model coincides with a serial connection of viscous elements. The viscosity of the Voigt part is given by constant c_1 and the viscosity of the damper part is given by c_2 , which depends on the force applied to the part. The resultant viscosity of the serial connection is then described by $c_1 c_2 / (c_1 + c_2)$. This resultant viscosity is mainly governed by the smaller viscosity. Namely, if $c_1 \ll c_2$, the resultant viscosity is almost equal to c_1 , and vice versa. Since the viscosity corresponding to the gravity force applied to a particle must take a large value, the corresponding force-dependent nonlinear damper should be dominant, say, $c_1 \gg c_{max}$. Consequently, we find that plastic objects can be described by the following condition:

$$k_1 = 0, \quad c_1 \gg c_{max}.$$

The viscosity of an object is then specified by c_{min} . Figure 11 shows an example of the modeling of a plastic object.

5.2 Physical Interaction among Rheological Objects

Let us describe the physical interaction among rheological objects in contact. Collisions between two rheological objects and collisions between two regions of a rheological object cause reaction forces at the contacting regions. We will apply the penalty method [18, 19] to compute the reaction forces. Namely, reaction forces can be simulated by introducing artificial forces similar to eq.(7). In a 2D model, we will define an artificial force between a vertex and an edge on the surfaces. When the signed distance between a vertex and an edge is below a threshold and the foot of the perpendicular of the vertex is on the edge, a force generated by a virtual Voigt model is applied to the vertex and to the edge in the opposite directions. In a 3D model, we will define an artificial force between a vertex and a triangle on the surfaces.

Figure 12 shows the collision of two regions of a virtual rheological object. An external force is exerted on the object at the center of its top face and is lost after a while. Two regions of the object are in contact with each other, as shown in Figure 12-(b). The two regions suffer no interference and the object deformation can be computed well, as shown in Figure 12-(c) and (d). Figure 13 shows the deformation of a rheological object pressed by a rigid object. A rigid object is put on a rheological object and is removed after a while. The rigid object is modeled as a viscoelastic object with large elasticity. The deformation process can be computed successfully as shown in the figure.

Let us simulate kinetic friction forces caused by the collision between two regions of rheological objects.

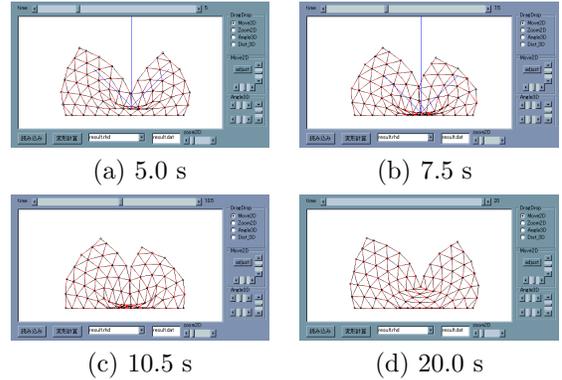


Figure 12: Self-collision of rheological object

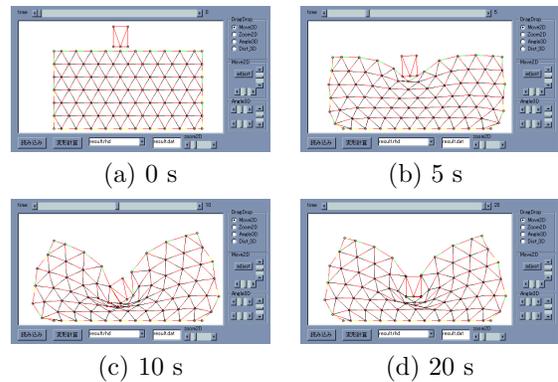


Figure 13: Deformation of rheological object pressed by rigid object

The magnitude of the kinetic friction can be computed by the Coulomb-Amontons law. The direction of the friction is determined by the relative velocity between a vertex and an edge in the 2D model or between a vertex and a triangle in the 3D model. Figure 14 demonstrates the effect of friction. In this demonstration, a small viscoelastic object is on a rheological object. Let us apply an external force to the right side of the rheological object. The small object then slides on the surface in the right direction. The coefficient of kinetic friction is equal to 0.0 in Figure 14-(a) while it is equal to 0.3 in Figure 14-(b). Since the surface is frictionless in Figure 14-(a) whereas it is frictional in Figure 14-(b), the small object slides more in Figure 14-(a) than it does in Figure 14-(b). As shown in the figure, we can simulate the kinetic friction forces between the two regions.

Figure 15 demonstrates the collision among six rheological objects under gravity. The proposed approach can simulate the physical interaction among multiple rheological objects, as shown in the figure.

6 Concluding Remarks

We have developed a systematic and coherent method to construct virtual rheological objects. First,

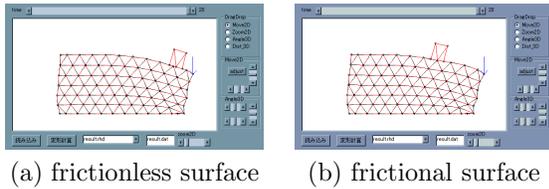


Figure 14: Description of kinetic friction

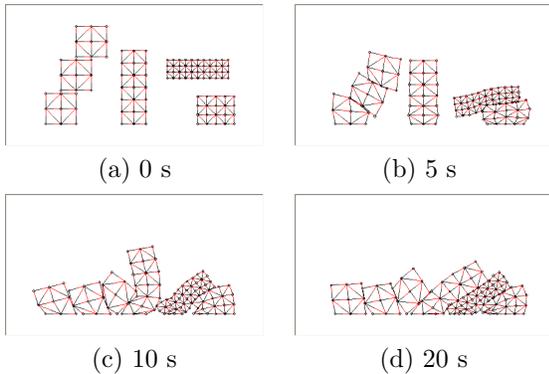


Figure 15: Collision among objects

we selected rheological elements appropriate for describing the deformation of rheological objects. It turned out that the three-element model with a force-dependent damper is appropriate. Secondly, we applied the discrete element approach to virtual rheological objects. We found that introducing a signed distance into a triangle/tetrahedron of a virtual rheological object and controlling the length ratio of the Voigt part are essential in order to keep the model topology consistent. Thirdly, we described the physical interaction among rheological objects. We have shown that our approach can describe viscoelastic, plastic, and rheological deformations in a systematic and coherent manner. It also turned out that the contact among rheological objects can be simulated appropriately.

Future issues include 1) identification of physical parameters in virtual rheological objects, 2) realtime computation of rheological deformation, and 3) dynamic simulation of food forming process and human mastication process.

References

- [1] Fung, Y. C., *Foundations of Solid Mechanics*, Prentice-Hall, 1965.
- [2] Barnes, H. A., Hutton, J. F., and Walters, K., *An Introduction to Rheology*, Elsevier Science Publishers, 1989.
- [3] Terzopoulos, D., Platt, J., Barr, A., and Fleischer, K., *Elastically Deformable Models*, Computer Graphics, Vol.21, No.4, pp.205–214, 1987.
- [4] Terzopoulos, D. and Witkin, A., *Deformable Models – Physically Based Models with Rigid and Deformable Components*, IEEE Computer

- Graphics and Applications, November, pp.41–51, 1988.
- [5] Zhuang, Y. and Canny, J., *Haptic Interaction of Global Deformations*, Proc. IEEE Int. Conf. Robotics and Automation, pp.2428–2433, 2000.
- [6] Picinbono, G., Delingette, H., and Ayache, N., *Non-linear and Anisotropic Elastic Soft Tissue Models for Medical Simulation*, Proc. IEEE Int. Conf. Robotics and Automation, pp.1371–1376, 2001.
- [7] Debunne, G., Desbrun, M., Cani, M.-P., and Barr, A. H., *Dynamic Real-Time Deformations using Space & Time Adaptive Sampling*, Computer Graphics Proceedings (SIGGRAPH 2001), pp.31–36, 2001.
- [8] James, D. L. and Pai, D. K., *ArtDefo – Accurate Real Time Deformable Objects*, Computer Graphics Proceedings (SIGGRAPH 1999), pp.65–72, 1999.
- [9] James, D. L. and Pai, D. K., *Real Time Simulation of Multizone Elastokinematic Models*, Proc. IEEE Int. Conf. Robotics and Automation, pp.927–932, 2002.
- [10] Eischen, J. W., Deng, S., and Clapp, T. G., *Finite-Element Modeling and Control of Flexible Fabric Parts*, IEEE Computer Graphics and Applications, pp.71–80, 1996.
- [11] Eberhardt, B., Weber, A., and Strasswer, W., *A Fast, Flexible, Particle-System Model for Cloth Draping*, IEEE Computer Graphics and Applications, pp.52–59, 1996.
- [12] Baraff, D. and Witkin, A., *Large Steps in Cloth Simulation*, Computer Graphics Proceedings (SIGGRAPH 1998), pp.43–54, 1998.
- [13] Terzopoulos, D., and Fleisher, K., *Modeling Inelastic Deformation: Viscoelasticity, Plasticity, Fracture*, Computer Graphics, Vol.22, No.4, pp.269–278, Alberquerque, May, 1988.
- [14] Galyean, T. A., and Hughes, J. F., *Sculpting: An Interactive Volumetric Modeling Technique*, Computer Graphics, Vol.25, No.4, pp.267–274, 1991.
- [15] Chai, Y., and Luecke, G. R., *Virtual Clay Modeling Using the ISU Exoskeleton*, Proc. IEEE Virtual Reality Annual Int. Symp., pp.76–80, 1998.
- [16] Witkin, A. and Welch, W., *Fast Animation and Control of Non-rigid Structures*, Computer Graphics Proceedings (SIGGRAPH 1990), pp.243–252, 1990.
- [17] Frugoli, G., Galimberti, A., Rizzi, C., and Bordegoni, M., *Discrete Element Approach for Non-rigid Material Modeling*, Henrich, D. and Wörn, H. eds., *Robot Manipulation of Deformable Objects*, Springer-Verlag, Advanced Manufacturing Series, pp.29–41, 2000.
- [18] Moore, M. and Wilhelms, J., *Collision Detection and Response for Computer Animation*, Computer Graphics, Vol. 22, No. 4, pp.289–298, 1988.
- [19] Joukhader, A., Deguet, A., and Laugie, C., *A Collision Model for Rigid and Deformable Bodies*, Proc. IEEE Int. Conf. on Robotics and Automation, pp.982–988, Leuven, May, 1998.

仮想レオロジー物体における接触のモデリング

杉山 勇太, 木村 政文, 友國 誠至, 平井 慎一
立命館大学ロボティクス学科

Modeling of Contact among Virtual Rheological Objects

Yuuta SUGIYAMA, Masafumi KIMURA, Seiji TOMOKUNI, and Shinichi HIRAI
Dept. Robotics, Ritsumeikan Univ.

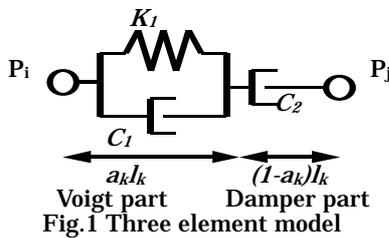
Abstract : In this paper, we will present a technique of modeling contact among virtual rheological objects. A collision of virtual rheological objects, which are modeled as a lattice structure model, is expressed using artificial forces between the edges and the nodes. We will develop the time-integrated penalty method, where an artificial force involves a time-integral term.

1. はじめに

食品や生体組織など、レオロジー的変形特性を有する物体に関しては、仮想物体を構築する手法が確立していない。レオロジー物体とは、戻り変位と残留変位を有する柔軟物である。著者らは、レオロジー物体をモデリングする手法を提案した[1][2]。さらに、物体同士の衝突や摩擦をモデリングする必要がある。衝突を表現する手法として、侵入量に比例した力を発生させるペナルティ法がある[3][4]。ペナルティ法は1回ステップあたりの計算量が少なく、並列リアルタイム計算に向いている。しかしペナルティ法では、物体がめり込んでいるように見える場合がある。本研究ではペナルティ法に侵入量の時間積分によって発生する力を追加した手法を提案する。

2. レオロジー要素

Fig.1に示す3要素モデルが、レオロジー物体の変形特性のモデリングに適している。要素の両端を P_i と P_j とする。フォークト部のバネ係数を K_1 、ダンパー係数を C_1 、フォークト部と直列に配置されたダンパーのダンパー係数を C_2 とする。稜線の長さを l_k 、稜線を分割する比率を a_k とすると、フォークト部の長さは $a_k l_k$ となる。このレオロジー要素を格子状に配置することによってレオロジー物体をモデリングする。



3. 接触のモデリング

3.1 提案するペナルティ法による衝突

Fig.2(a)に示す物体Aと物体Bの衝突を、局所的な稜線と頂点の関係に基づいて表現するとFig.2(b)になる。ペナルティ法では干渉が生じた時に、干渉を解消する人工力を発生させる。稜線Qと頂点Rの垂直方向の長さを L 、垂直方向の相対速度を V とする。従来のペナルティ法では、 L の値を負とし、頂点Rに与える人工力 F_c は、

$$F_c = -K_c L - C_c V \quad (1)$$

と表される。ここで、 K_c はバネ係数、 C_c はダンパー係数を表す。従来のペナルティ法では、重力や電磁気力等の物体力 F_g により干渉が生じた場合、干渉は完全には解消されない。例えば、物体を床に置いた場合に、物体が床にのめりこんだままとなる。この場合は、重力と釣り合うための人工力 F_c を発生させるために、侵入量 L が必要だからである。

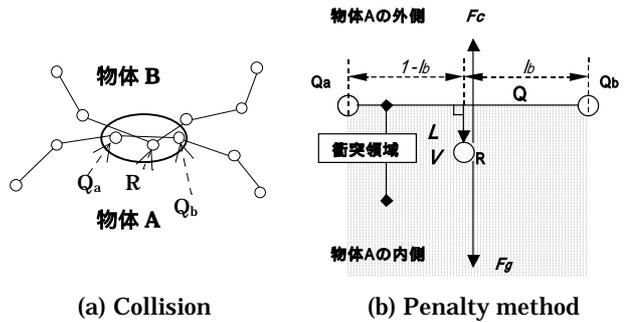


Fig.2 Collision between node and edge

そこで、時間が経つにつれて理想的な状態である、頂点が稜線上にある状態に近づけるために、ペナルティ法に侵入量の時間積分の項を加える。以下この手法をペナルティ時間積分法と呼ぶ。Fig.3に示すペナルティ時間積分法での人工力 F_{co} は、

$$F_{co} = -K_c L - C_c V - I_c \int_{t_0}^{t_1} L dt \quad (2)$$

と表される。ここで、 I_c は積分係数、 t_0 は衝突判定領域に侵入した時間、 t_1 は現在の時間とする。

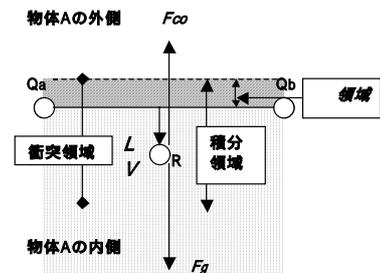


Fig.3 Time-integrated penalty method

時間積分の値は、衝突判定領域から離れると、初期値0に戻るものとする。離散的な計算では、稜線上での人工力 F_{co}

と物体力 F_g は必ずしも等しくならなく, $F_{co}>F_g$ の場合では頂点は稜線の外側へ離れてしまう. そうなると, 時間積分の値が初期値に戻ってしまうため, 頂点Rは振動的な挙動を示す. そこで, 衝突判定領域を微小な長さ だけ稜線の外側に設け, その間に頂点がある場合でも時間積分をそのまま続けるようにする. その領域を 領域とする. これにより頂点Rは稜線近傍に収束する.

3.2 安定性の向上

Fig.4は物体の鋭角な部分に頂点Pが稜線Mから, 物体内に入った場合であり, 稜線M,Nからの侵入量を L_m, L_n とすると,

$$-L_m < -L_n \quad (3)$$

となる. このような干渉を(2)式のバネ力の項のみで表すと, 頂点Pに稜線M, Nより発生する人工力 F_m, F_n は,

$$\begin{aligned} F_m &= -K_c L_m \\ F_n &= -K_c L_n \end{aligned} \quad (4)$$

となる. この時, それぞれのバネ係数は,

$$\frac{F_m}{-L_m} = \frac{F_n}{-L_n} = K_c \quad (5)$$

となる. ここで, 頂点Pはなるべく稜線M側よりに出るようにしたい. そこで, (2)式より導かれる, 頂点と稜線の間に発生する人工力 F_{co} に, 定常的な力 K_c を加える. 新たな人工力 F_c は,

$$F_c = F_{co} + K_c \quad (6)$$

となる. (5)式のようにバネ係数を算出すると,

$$K_c + \frac{K_c}{-L_m} = \frac{F_m}{-L_m} > \frac{F_n}{-L_n} = K_c + \frac{K_c}{-L_n} \quad (7)$$

となる. これは, 侵入量 L が短いほど, みかけのバネ係数が大きいことを示している. よって, 頂点Pは稜線M側よりに出やすくなる. (6)式をペナルティ時間積分法の形で表すと, 人工力 F_d は,

$$F_c = -K_c(L -) - C_c V - I_c \int_{t_0}^{t_1} L dt \quad (8)$$

ただし,

$$\int_{t_0}^{t_1} L dt \leq 0$$

となる. つまり, 領域の始めをバネの初期位置とする. (8)式により, 頂点は, 物体力 $F_g \geq K_c$ の時には $L=0$ に収束し, $F_g < K_c$ の時には 領域内で収まる.

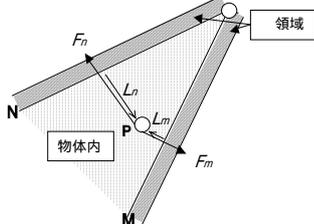


Fig.4 Collision with acute region of object

3.3 力の分配法

稜線を構成する頂点には, 衝突によって発生する力の反作用を与える必要がある. Fig.2(b)に示すように, 稜線を構成する頂点を Q_a, Q_b とし, 頂点Rの稜線 $Q_a Q_b$ への足が稜線を内分する比を $(1-l_b):l_b$ とする. この時, Q_a, Q_b に作用する力 F_{ac}, F_{bc} は,

$$\begin{aligned} F_{ac} &= -l_b F_c \\ F_{bc} &= -(1-l_b) F_c \end{aligned} \quad (9)$$

となる.

3.4 摩擦力

衝突により発生する力 F_c を垂直抗力とみなすことにより, クーロン・アモントン則により, 摩擦力を表現できる.

4. 実験

従来のペナルティ法と, ペナルティ時間積分法を用いた場合による物体の位置比較を行う. 床の高さを0として物体を床の上に置いた場合の物体の位置を示す.

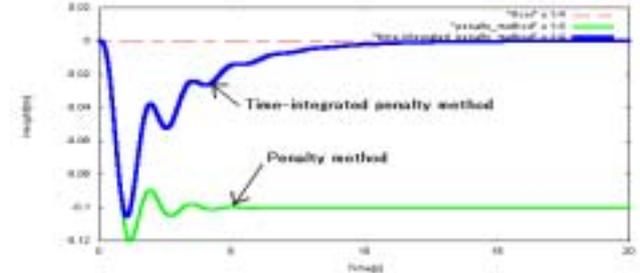
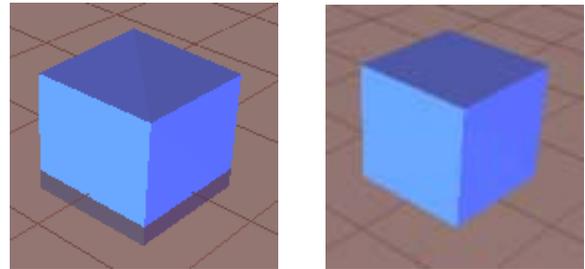


Fig.5 Height of object



(a) Penalty method (b) Time-integrated penalty method

Fig.6 Height of object in 3D space

Fig.5のように, 従来のペナルティ法では物体が床に侵入した状態が続くが, ペナルティ時間積分法では最終的には高さ0の位置になる. Fig.6はペナルティ時間積分法を3次元に応用した結果である. 3次元では面と点, 稜線と稜線の関係に基づいてペナルティ時間積分法を適用すれば, 2次元と同様の結果が得られる.

5. まとめ

ペナルティ時間積分法を用いると, 物体が侵入しつづけることなく, 稜線上で釣り合うことを示した. 今後, 実物体の接触から, ペナルティ時間積分法のバネ係数, ダンパー係数, 積分係数を同定する手法を確立する.

参考文献

- [1] 徳本, 平井, 形状制御のためのレオロジー物体の変形モデリング, 日本ロボット学会誌, Vol.18, No.8, pp.1141-1148, 2000
- [2] 木村, 杉山, 友國, 平井, 仮想レオロジー物体の変形シミュレーション, 日本バーチャリアリティ学会第7回大会論文集, pp.183-186, 2002
- [3] Terzopoulos.D, Platt.J, Barr.A, and Fleisher.K, *Elastically Deformable Models*, Computer Graphics, Vol.21, No.4, pp.205-214, 1987
- [4] Moore.M, Wilhelms.J, *Collision Detection and Response for Computer Animation*, Computer Graphics, Vol.22, No.4, pp.289-298, August 1988

三要素モデルによる粘弾性物体，塑性物体およびレオロジー物体のモデリング

友國 誠至，木村 政文，杉山 勇太，平井 慎一（立命館大学）

Modeling of viscoelastic objects, plastic objects, and rheological objects using three element models

Seiji Tomokuni, Masafumi Kimura, Yuuta Sugiyama, and Shinichi Hirai

Ritsumeikan University

Abstract: In this report, we will present a method to construct the physical models of viscoelastic objects, plastic objects, and rheological objects using three element models in a coherent manner. We introduce constraints into the ratio of voigt part. Viscoelastic objects, plastic objects, and rheological objects are modeled by setting the parameter of the three element models with constraints.

Keywords: Modeling, Deformation, Virtual

1. はじめに

これまで，現実世界に存在する様々な物体を仮想空間に表現する事を目的とした研究が精力的に進められてきた．特に，仮想的な粘弾性物体および塑性物体の表現手法については多くの研究がなされており，また，我々のグループは食品や生体組織に代表されるレオロジー物体を仮想空間に構築する手法を提案した[1]．本報告では，レオロジー的性質を持つ三要素モデルを用いて，粘弾性物体，塑性物体，レオロジー物体を統一的に扱う手法を提案する．

2. 各物体の性質

Fig.1(a)に示す形状を有する物体に外力を作用させ，Fig.1(b)の状態に変形させたとする．外力を解放すると粘弾性物体では，Fig.1(c)に示すように元の形状に戻る．塑性物体の場合は Fig.1(d)のように変形したままの形状を保つ．レオロジー物体の場合は Fig.1(e)のように，残留変位と戻り変位を生じる．

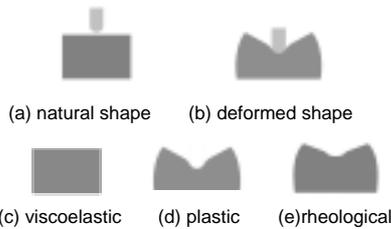


Fig.1 Viscoelastic object, plastic object, and rheological object

3. レオロジー物体の表現

本報告では，戻り変位および残留変位を表すモデルとして，Fig.2に示す三要素モデルを用いる．物体は三角形要素あるいは四面体要素を組み合わせることで表現する．各要素の頂点に質点を配し，頂点間を結ぶ稜線を三要素モデルとすることで二次元形状および三次元形状を表現する．

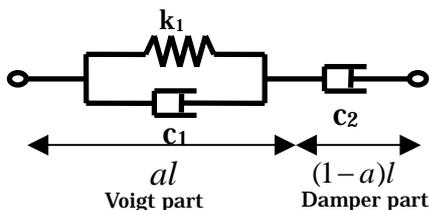


Fig.2 Three element model

次に，三要素モデルを定式化する．三要素モデルのフォークト部の比率を a とし，フォークト部の粘性係数，弾性係数，自然長を c_1, k_1, L とする．ダンパー部の粘性係数を c_2 とする．フォークト部に作用する力は次式で表される．

$$f = -K(al - L) - c_1(\dot{x} + a\ddot{x}) \quad (1)$$

同様に，ダンパー部に作用する力は次式で表される．

$$f = -c_2(-\dot{x} + (1-a)\ddot{x}) \quad (2)$$

フォークト部とダンパー部に作用する力は等しいため，(1)式および(2)式から，次式が導かれる．

$$\dot{x} = \frac{-K(al - L) - (c_1a - c_2(1-a))\ddot{x}}{(c_1 + c_2)l} \quad (3)$$

ここで，変数 a は以下の条件を満たさなくてはならない．

$$0 \leq a \leq 1 \quad (4)$$

しかし，(3)式を元に計算を行う限り，変数 a は計算過程でこの条件を逸脱する可能性がある．そこで，本報告では，変数 a に以下の制約条件を課す．

$$a_{\min} \leq a \leq a_{\max} \quad (5)$$

ここで a_{\min} ， a_{\max} は定数である．計算過程で変数 a の値を監視し，値が a_{\min} を下回った時には変数 a に a_{\min} を代入し，変数 a の値が a_{\max} を上回った場合には，変数 a に a_{\max} を代入する．

以上を元に，各質点に接続する複数の三要素モデルによって生じる力を合計し，運動方程式に基づいて各質点に加速度の計算する．質点の位置，速度，フォークト部の比率をルンゲクッタ法による数値計算で求めることで，物体の変形シミュレーションを行う．

構築したレオロジー物体の変形シミュレーションを Fig.3に示す．

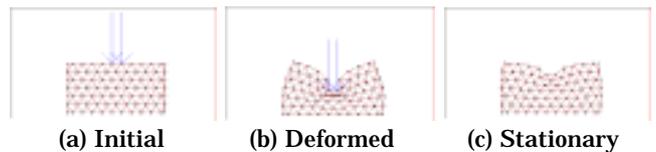


Fig.3 Simulation of rheological object

4. 粘弾性物体の表現

粘弾性は、フォークトモデルによって表現することが可能である。三要素モデルは、ダンパー部の粘性係数 c_2 の値を十分に大きな値に設定することで、フォークトモデルと同様の挙動を示す。長時間の変形シミュレーションを行う場合、ダンパー部の変形量が蓄積される可能性があるが、 a_{min} , a_{max} とともに 1 に設定することでこれを回避できる。この手法で構築した粘弾性物体の変形シミュレーションを Fig.4 に示す。

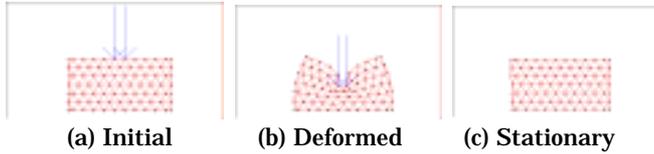


Fig.4 Simulation of viscoelastic object

5. 塑性物体の表現

塑性は、単一のダンパーによって表現することが可能である。三要素モデルは、フォークト部の弾性係数 k の値を 0 に設定することで、直列に接続した二つのダンパーとなる。この場合、三要素モデル全体の粘性係数 c は次式で表される。

$$c = \frac{c_1 c_2}{c_1 + c_2} \quad (6)$$

粘性係数として c_{damper} の値をもつダンパーを表現するには、 c_1, c_2 の値を共に c_{damper} の 2 倍に設定すればよい。この手法で構築した塑性物体の変形シミュレーションを、Fig.5 に示す。

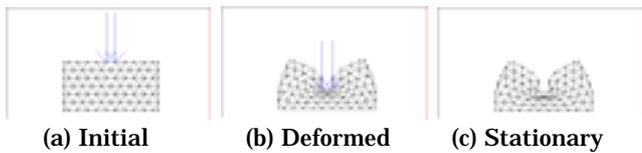


Fig.5 Simulation of plastic object

6. 力依存型ダンパー

本報告における三要素モデルは単独のダンパーを含むため、仮想空間に重力を導入すると、レオロジー物体および塑性物体の形状は時間の経過とともに崩れてしまう。そこで、弱い力に対して選択的に強い粘性を示す力依存型ダンパーを三要素モデルのダンパー部に導入することにより、重力下で形状を保つ物体を構築する事ができる。力依存型ダンパーの粘性係数 c_2 を以下の式で与える。

$$c_2 = \begin{cases} c_{max} & (f \leq f_1) \\ Ae^{-Bf} & (f_1 \leq f \leq f_2) \\ c_{min} & (f \geq f_2) \end{cases} \quad (7)$$

ここで $A, B, f_1, f_2, c_{max}, c_{min}$ は定数であり、これらは(7)式における粘性係数 c_2 の値が連続となるように設定する。

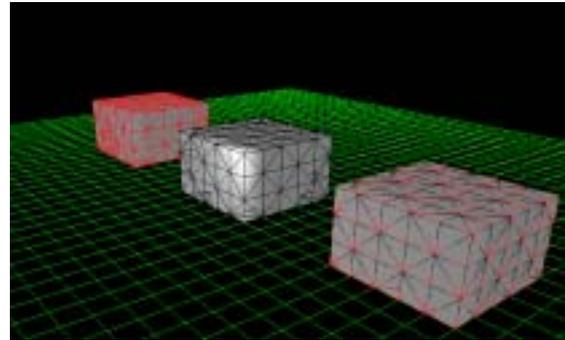
力依存型ダンパーを用いた三要素モデルを塑性物体に対して適用する場合、線形ダンパーと力依存型ダンパーが直列に連結した形となる。粘性係数 c_1 の値が c_2 よりも十分に大きな値であれば、全体の粘性係数は次式で近似できる。

$$\frac{c_1 c_2}{c_1 + c_2} \approx c_2 \quad (8)$$

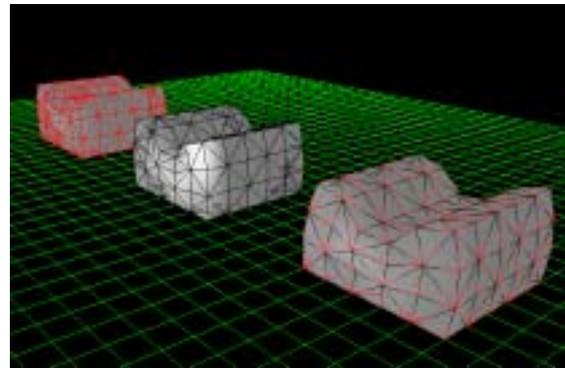
これにより力依存型ダンパーを導入した三要素モデルにおいて、単体の力依存型ダンパーと同等の挙動を表現する事が可能である。

7. 三次元物体

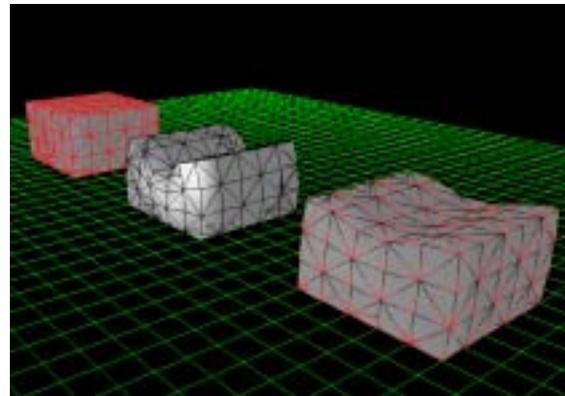
本報告で提案した手法は三次元物体にも適用できる。Fig.6 は、三次元空間における粘弾性物体、塑性物体、レオロジー物体の変形シミュレーションである。物体上面部を変形させることで物体側面部が膨らんでいることが確認できる。



(a) Initial



(b) Deformed



(c) Stationary

Fig.6 Simulation of 3D objects

8. おわりに

本報告では、粘弾性物体、塑性物体およびレオロジー物体を統一的にモデリングする手法を提案した。我々は現在、レオロジー物体の高速変形計算を目的とし、FPGA を用いた三要素モデルの並列計算回路の開発を進めている。今後は、今回提案した手法を用いることで、同一の計算回路による粘弾性物体および塑性物体の高速変形計算を目指す。

参考文献

- 1) 徳本真一, 平井慎一, 形状制御のためのレオロジー物体の変形モデリング, 日本ロボット学会誌, Vol.18, No.8, pp.87-94, 2001

仮想レオロジー物体の変形シミュレーション

Deformation Simulation in Virtual Rheological Objects

木村政文, 杉山勇太, 友國誠至, 平井慎一

Masafumi KIMURA, Yuuta SUGIYAMA, Seiji TOMOKUNI, and Shinichi HIRAI

立命館大学ロボティクス学科

(〒 525-8577 滋賀県草津市, hirai@se.ritsumei.ac.jp)

Abstract : A physical modeling of virtual rheological objects is presented. Objects showing rheological nature involve food and biological tissues while no systematic approach to build their virtual objects can be found. In this report, we will construct a physical model of 2D/3D rheological objects.

Key Words: rheological objects, modeling, deformation, contact, friction

1. はじめに

仮想空間を構築するときには, 現実世界に存在する様々な特性を有する物体を構築する必要がある. 粘弾性物体や塑性物体に関しては, 仮想物体を構築する手法が提案されている [1, 2]. 一方, 食品や生体組織など, レオロジー的変形特性を有する物体に関しては, 仮想物体を構築する手法は確立していない. レオロジー物体とは, 戻り変位と残留変位を有する柔軟物であり, 多様な変形特性を示す. 本報告では, 仮想レオロジー物体を構築するために, レオロジー物体のモデリング手法と変形計算手法を提案する.

2. レオロジー物体

図 1-(a) に示す初期形状を有する物体に外力を作用させると, 図 1-(b) に示すように変形すると仮定する. 粘弾性物体では, 図 1-(c) に示すように, 外力を解放したときの形状が初期形状に一致する. すなわち, 戻り変位があり, 残留変位はない. 塑性物体では, 図 1-(d) に示すように, 外力を解放したときの形状が変形形状に一致する. すなわち, 残留変位があり, 戻り変位はない. 図 1-(e) に示すように, 戻り変位と残留変位の両方を有する物体を, レオロジー物体とよぶ. 食品や生体組織は, レオロジー物体に分類される.

三角形要素あるいは四面体要素を組み合わせると, 二次元あるいは三次元形状を表す. 各要素の頂点に質点を, 稜線にレオロジー要素を配置することにより, レオロジー物体の変形過程を表すことができる. したがって, 物体モデルは, 質点の集合と質点を結ぶレオロジー要素の集合から構成される. 質点に順次番号を付し, 第 i 質点を P_i で表す. また, レオロジー要素に順次番号を付し, 第 k 要素を E_k で表す. レオロジー要素 E_k は, 両端の質点番号を属性として含む. 両端の一方をレオロジー要素の始点, 他方を終点とよぶ.

仮想レオロジー物体の運動方程式を導く. 図 2 に示すように, 質点 P_i と P_j が三要素モデル E_k で接続されている. ただし, 始点を P_i , 終点を P_j とする. 質点

P_i の位置を x_i , 速度を v_i , 質量を m_i で表す. レオロジー要素の長さを l_k とし, フォークト部の長さを $a_k l_k$ とする. このとき, ダンパー部の長さは $(1 - a_k) l_k$ で与えられる. また, フォークト部の自然長を L_k で表す. 図 2 に示す力学系の状態変数は, x_i, v_i, x_j, v_j, a_k である. フォークト部の伸びは $a_k l_k - L_k$ に等しいので, 三要素モデル E_k がフォークト部に与える力の大きさを f_k は,

$$f_k = -k_1 \{a_k l_k - L_k\} - c_1 \{\dot{a}_k l_k + a_k \dot{l}_k\} \quad (1)$$

である. ダンパー部の長さは $(1 - a_k) l_k$ に等しいので, 三要素モデルがダンパー部に与える力は,

$$f_k = -c_2 \{-\dot{a}_k l_k + (1 - a_k) \dot{l}_k\} \quad (2)$$

である. フォークト部に作用する力とダンパー部に作用する力は等しいので,

$$\dot{a}_k = \frac{-k_1 \{a_k l_k - L_k\} - \{c_1 a_k - c_2 (1 - a_k)\} \dot{l}_k}{(c_1 + c_2) l_k} \quad (3)$$

が得られる. レオロジー部の長さ l_k は, 次式を満たす.

$$l_k^2 = (\mathbf{x}_i - \mathbf{x}_j) \cdot (\mathbf{x}_i - \mathbf{x}_j). \quad (4)$$

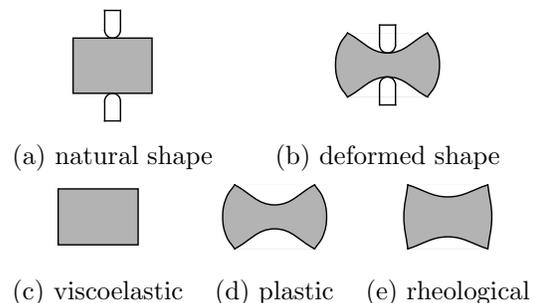


図 1: 粘弾性物体, 塑性物体, レオロジー物体

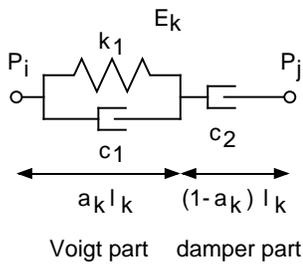


図 2: 三要素モデル

上式を時間微分すると

$$\dot{l}_k = \frac{(\mathbf{x}_i - \mathbf{x}_j) \cdot (\mathbf{v}_i - \mathbf{v}_j)}{l_k}. \quad (5)$$

したがって、三要素モデル E_k が始点 P_i に与える力は $f_k e_k$ であり、終点 P_j に与える力は $-f_k e_k$ である。ここで、 e_k は、始点から終点に向かう単位ベクトルであり、次式で与えられる。

$$e_k = \frac{\mathbf{x}_j - \mathbf{x}_i}{l_k}. \quad (6)$$

質点 P_i を始点とする三要素モデルの集合を R_i 、質点 P_j を終点とする三要素モデルの集合を S_j で表す。このとき、集合 R_i に含まれる三要素モデル E_k が、質点 P_i に加える力は $f_k e_k$ に一致する。また、集合 S_j に含まれる三要素モデル E_k が、質点 P_j に加える力は $-f_k e_k$ に一致する。したがって、質点 P_i の運動方程式は、

$$m_i \dot{\mathbf{v}}_i = \sum_{k \in R_i} f_k e_k - \sum_{k \in S_i} f_k e_k + \mathbf{F}_i^{ext} \quad (7)$$

と表される。ここで \mathbf{F}_i^{ext} は、質点 P_i に作用する外力である。結局、物体モデルの運動方程式は、(3)(7) 式で与えられる。したがって、運動方程式の解を数値的に計算することにより、仮想レオロジー物体の変形を求めることができる。

3. 位相保持

物体モデルにおいて、変形を正確に計算するためには、質点どうしの位相的な接続関係が保たれなくてはならない。しかしながら、離散時間で変形を計算するときには、位相的な接続関係が崩れ、変形計算に失敗することがたびたび生じる。そこで本節では、物体モデルにおいて、位相的な接続関係を保つ手法を検討する。

二次元物体モデルにおいて位相的な接続関係が崩れるのは、図 3-(a),(b) に示すような鏡像どうしを区別できないことに起因する。そこで、鏡像どうしが区別できるように、個々の稜線に向きを付ける。稜線に向きが定義されているので、図 3 に示すように、三角要素の格子点 P_k と稜線 $P_i P_j$ との間に、符号付距離を定義することができる。稜線 $P_i P_j$ に垂直で、自然状態における P_k を向く単位ベクトルを \mathbf{n}_{ij}^k 、稜線 $P_i P_j$ と格子点 P_k との符号付き距離を d_{ij}^k とする。位相的な接続関係を保つためには、符号付き距離 d_{ij}^k の値が小さくなったとき、質点 P_k に稜線から離れる方向の力を作用させればよい。本報告では、このような力が

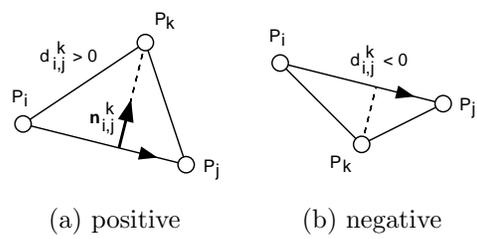


図 3: 頂点と稜線の符号付き距離

仮想的なフォークト要素により生成されると考え、以下に示す人工的な力を導入する。

$$\mathbf{f}_{ij}^k = \begin{cases} \mathbf{0} & (d_{ij}^k > \epsilon) \\ \{-K(d_{ij}^k - \epsilon) - C\dot{d}_{ij}^k\} \mathbf{n}_{ij}^k & (d_{ij}^k \leq \epsilon) \end{cases} \quad (8)$$

ここで、 K, C はフォークト部の弾性係数、粘性係数、 ϵ は微小な正の定数である。符号付き距離 d_{ij}^k が閾値 ϵ より小さくなったとき、仮想的なフォークト要素が生成する力により、質点 P_k は稜線 $P_i P_j$ より離れる方向に動く。なお、三次元物体モデルにおいては、四面体要素の頂点と対向する三角形との間で、同様の符号付き距離が定義でき、上述の議論を適用することができる。

位相的な接続関係が崩れる原因で、レオロジー要素特有の問題として、フォークト部とダンパー部の長さの比率がある。フォークト部の長さの比率は、 a_k で与えられる。変数 a_k は、条件 $0 \leq a_k \leq 1$ を満たさなくてはならない。しかしながら、計算過程において、この条件が満たされなくなり、位相的な接続関係がたびたび崩れる。そこで本報告では、フォークト部の長さ比率の許される最小値を a_{min} 、最大値を a_{max} で表し、変数 a_k に対して制約条件

$$a_{min} \leq a_k \leq a_{max} \quad (9)$$

を課す。計算過程で a_k の値が a_{min} を下回ったときには、 a_k の値を a_{min} とする。また、 a_k の値が a_{max} を越えたときには、 a_k の値を a_{max} とする。

4. 力依存型非線形ダンパー

レオロジー要素は、残留変位を表すために、独立の粘性要素を含んでいる。通常の三要素モデルでは、独立の粘性要素を線形ダンパーで表す。しかしながら、

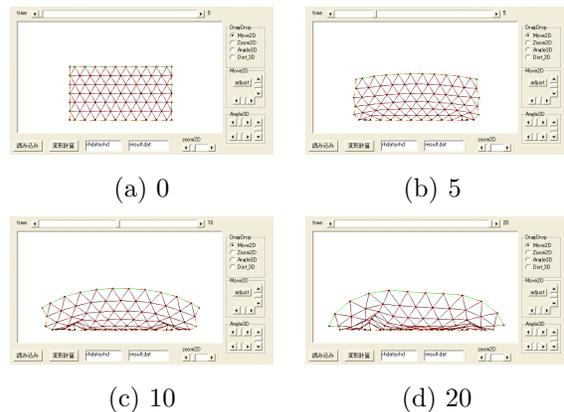
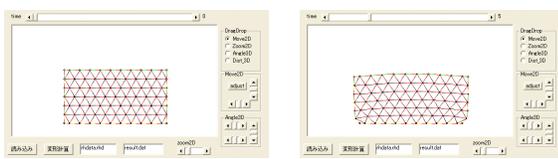
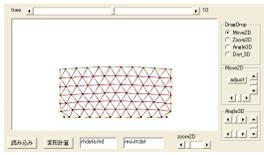


図 4: 線形ダンパー

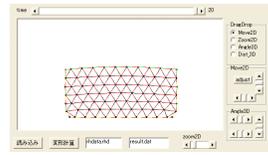


(a) 0

(b) 5

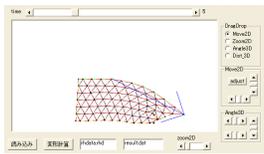


(c) 10



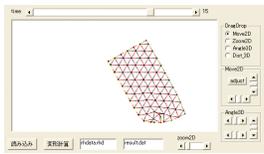
(d) 20

図 5: 力依存型非線形ダンパー

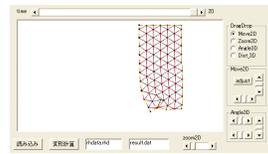


(a) 5

(b) 10

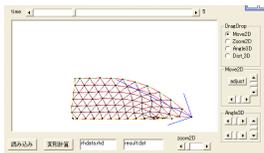


(c) 15



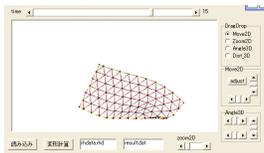
(d) 20

図 6: 重力なし

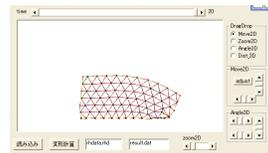


(a) 5

(b) 10



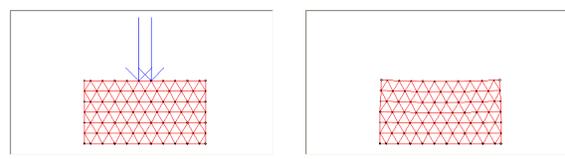
(c) 15



(d) 20

図 7: 重力あり

線形ダンパーにおいては、力が作用し続ける限り、変位が増加あるいは減少し続ける。この性質は、重力を導入する障害になる。すなわち、質点に重力が作用し続ける限り、レオロジー要素が変形し、結果として物体の形状が崩れてしまう。図 4 に、線形ダンパーを含む三要素モデルから構成される物体モデルに対して、変形過程を計算した例を示す。図に示すように、重力が作用している限り物体が変形し、形状が崩れる。そこで、本報告では、独立の粘性要素に、力依存型非線形ダンパーを導入する。力依存型非線形ダンパーでは、ダンパーに作用する力の大きさ f に応じて、ダンパーの粘性係数が変化する。重力下で物体の形状を保つためには、ダンパーに作用する力が小さいときに、ダン



(a) initial

(b) final

図 8: 粘弾性物体のモデリング



(a) initial

(b) final

図 9: 塑性物体のモデリング

パーの粘性係数を大きくし、変形が進まないようにすればよい。本報告では、力依存型非線形ダンパーの粘性係数を、次式で与える。

$$c_2 = \begin{cases} c_{max} & (f \leq f_1) \\ Ae^{-Bf} & (f_1 \leq f \leq f_2) \\ c_{min} & (f \geq f_2) \end{cases}$$

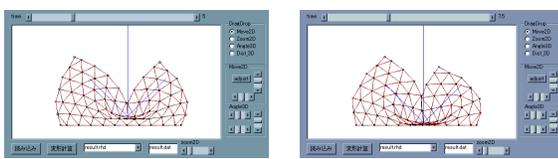
ここで、 f_1, f_2, A, B は定数であり、 $c_{max} = Ae^{-Bf_1}$ 、 $c_{min} = Ae^{-Bf_2}$ を満たす。力依存型非線形ダンパーを含む三要素モデルから構成される物体モデルに対して、変形過程を計算した例を図 5 に示す。図より、重力下で変形形状が安定に保たれていることがわかる。

力依存型非線形ダンパーを導入することにより、重力に代表される物体力の有無にかかわらず、変形形状を安定に計算することができる。図 6、図 7 は、テーブル上のレオロジー物体に外力を加えたときの挙動を表す。図 6 に示すように、重力が無い場合は、外力が 0 になると物体とテーブル間に作用する反力でレオロジー物体が上方に動く。一方、図 7 に示すように、重力が作用している場合は、テーブルから離れた部分が再びテーブルと接触する。

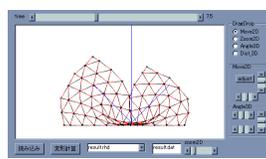
5. 粘弾性と塑性の表現

本報告で提案する手法を用いると、粘弾性物体から塑性物体までを、統一的に表現することができる。粘弾性は、レオロジー要素のフォークト部のみでモデリングできる。フォークト部の比率 a_k には、(9) 式に示す制約が加えられている。したがって粘弾性は、 a_{min} と a_{max} の値を、ともに 1 に設定することにより表すことができる。粘弾性物体のモデリング例を、図 8 に示す。

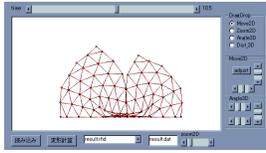
塑性は、単一のダンパーで表すことができる。フォークト部の弾性係数 k_1 の値を 0 に設定することにより、三要素モデルは直列接続された二つのダンパーに一致する。フォークト部の粘性係数 c_1 と非線形ダンパーの粘性係数 c_2 が直列に接続された場合、全体の粘性係数は、 $c_1 c_2 / (c_1 + c_2)$ で与えられる。ただし、重力下で物体の形状を保つためには、ダンパーに作用する力が小さいときに、非線形ダンパーの挙動が支配的であることが望まれる。すなわち、 $c_1 \gg c_2$ が満たされなくてはならない。また、この条件が満たされると



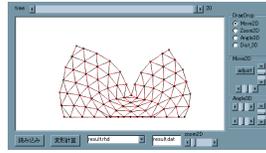
(a) 5



(b) 7.5



(c) 10.5



(d) 20

図 10: 二つの部分の衝突

きには、ダンパーに作用する力の大小にかかわらず、非線形ダンパーの挙動が支配的である。以上の議論より、塑性を表現するときには、

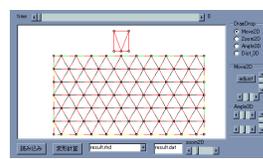
$$k_1 = 0, \quad c_1 = c_{max}$$

と設定すればよい。塑性の特性は、非線形ダンパーのパラメータ c_{min} と f_2 の値を指定することにより表す。塑性物体のモデリング例を、図 9 に示す。

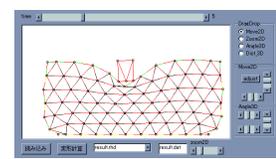
6. 衝突と摩擦の表現

レオロジー物体と他のレオロジー物体との衝突、あるいはレオロジー物体の二つの部分の衝突が生じると、衝突部分に抗力が発生する。表面の頂点と稜線あるいは表面の頂点と三角形の間で、(8) 式と類似の人工力を定めることより、抗力を表現することができる。二次元モデルの場合、頂点と稜線の距離がある閾値以下であり、かつ頂点から稜線に降ろした足が稜線上にある場合、頂点と稜線にフォークト要素による力を作用させる。三次元モデルの場合、頂点と三角形の間で、同様の計算を行う。図 10 に、衝突を考慮して物体変形を計算した例を示す。図 10-(b) に示すように、力の作用点近傍で、物体の左側部分と右側部分が衝突している。しかしながら、図 10-(c),(d) に示すように、二つの部分が干渉することなく、変形が計算できていることがわかる。また、図 11 に、レオロジー物体上に剛体を置いたときの挙動を計算した例を示す。剛体は、弾性係数の値が大きい粘弾性物体として表す。これにより、二つのレオロジー物体の衝突として、挙動を計算することができる。図に示すように、物体の変形が計算されている。

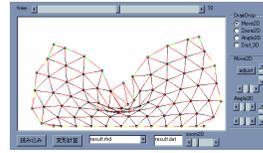
物体どうしの衝突に起因する抗力を求めているので、動摩擦を計算することができる。動摩擦力の大きさは、クーロン・アモントン則から計算する。動摩擦力の方向は、頂点と稜線あるいは頂点と三角形の相対速度より決定する。図 12 に、動摩擦の有無による挙動の違いを示す。この例では、大きいレオロジー物体の上に小さい粘弾性物体を載せ、レオロジー物体の右側に外力を下向きに作用させる。すると、粘弾性物体が右向きに滑る。粘弾性物体は、レオロジー物体としてモデリングする。図 12(a) では、表面に摩擦が作用しないので、粘弾性物体は、大きく滑る。一方、図 12(b) では、表面に摩擦が作用するので、粘弾性物体は、少ししか滑べない。動摩擦係数は 0.3 と設定した。このように、レオロジー物体間あるいはレオロジー物体の部分間の動摩擦を表現することができる。



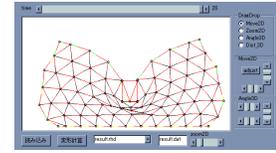
(a) 0



(b) 5

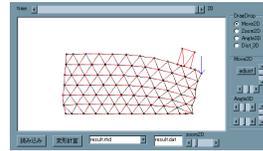


(c) 10

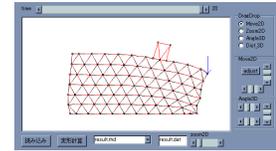


(d) 20

図 11: レオロジー物体上に剛体を置いたときの挙動

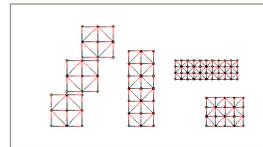


(a) frictionless surface

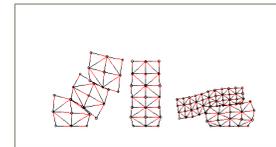


(b) frictional surface

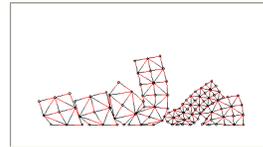
図 12: 動摩擦の有無による挙動の違い



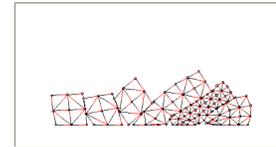
(a) 0



(b) 5



(a) 10



(b) 20

図 13: 複数物体の衝突

提案する手法を用いると、図 13 に示すように、複数のレオロジー物体の衝突を表現することができる。

7. おわりに

本報告では、レオロジー物体のモデリング手法と変形計算手法を提案した。提案する手法は、すでに三次元に拡張した。今後は、1) 実物体からのモデル同定、2) FPGA を用いた高速計算、3) 力覚提示装置との接続を進める。

参考文献

- [1] Terzopoulos, D., Platt, J., Barr, A., and Fleisher, K., *Elastically Deformable Models*, Computer Graphics, Vol.21, No.4, pp.205-214, 1987
- [2] Chai, Y., and Luecke, G. R., *Virtual Clay Modeling Using the ISU Exoskeleton*, Proc. IEEE Virtual Reality Annual Int. Symp., pp.76-80, 1998

Chapter 2

Particle-based Modeling of Object Deformation

This chapter introduces *particle-based modeling*, which can describe 2D/3D object deformation in a simple and intuitive manner. This particle-based approach has developed in computer graphics [4, 5] to simulate 2D/3D object deformation. In the particle-based modeling, an object is described by a set of finite number of particles. The motion and the deformation of the object can be sketched by the motion of individual particles. Specifying internal forces among the particles characterizes the deformation of the object. Solving motion equations of particles provides the motion and the deformation of the object.

Section 2.1 describes deformation elements to formulate 1D object deformation. Elasticity and viscosity are briefly reviewed. Parallel and serial connection among elastic and viscous elements are then introduced to formulate viscoelastic, viscoplastic, and rheological deformations. Section 2.2 and 2.3 introduce particle-based modeling of 2D/3D object deformation.

2.1 Deformation Elements

2.1.1 Elasticity

Elasticity determines the relationship between force and displacement. We can feel the hardness of a deformable object by pushing the object surface by a finger and observing the surface displacement against the applied force.

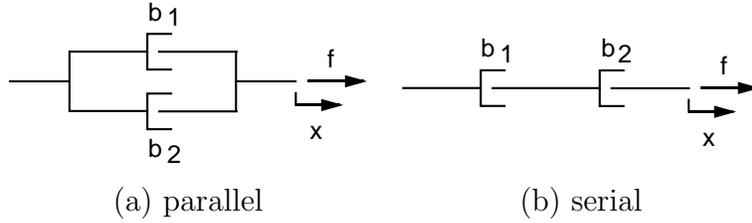


Figure 2.2: Connection of viscous elements

2.1.2 Viscosity

Viscosity determines the relationship between force and the rate of displacement. We can examine our skin condition by pushing a skin by a finger before releasing the pushing finger and observing the motion of skin surface after the release. Surface of a healthy skin goes back to its natural position quickly while surface of a swollen skin may go back slowly. This rate of displacement comes from the viscosity of the skin. Generally, the relationship is nonlinear and time-variant as well. We often assume an ideal relationship: *linear and time-invariant viscosity*, which is represented by a linear viscous element. Let d be the extension of the element and f be the generated force along its extension. Linear elastic element is then described as

$$f = b\dot{d}$$

where constant b is referred to as *viscous coefficient*.

Let us investigate the force-extension relationship in a system of two viscous elements connected in parallel illustrated in Figure 2.2-(a). Two viscous forces generated by the elements are given by $f_1 = b_1\dot{x}$ and $f_2 = b_2\dot{x}$. Thus, coefficient b of a viscous element equivalent to the two viscous elements connected in parallel is given by

$$b = b_1 + b_2.$$

Note that viscous coefficient b is larger than b_1 and b_2 . Let us investigate the force-extension relationship in a system of two viscous elements connected in serial illustrated in Figure 2.2-(b). Two extensions of the viscous elements satisfy $f = b_1\dot{x}_1$ and $f = b_2\dot{x}_2$. Thus, coefficient b of a viscous element equivalent to the two viscous elements connected in serial is given by the following equation:

$$\frac{1}{b} = \frac{1}{b_1} + \frac{1}{b_2}.$$

Note that viscous coefficient b is smaller than b_1 and b_2 .

2.1.3 Voigt model

Voigt model describes viscoelastic deformation of an object. Voigt model consists of an elastic element and a viscous element connected in parallel, as illustrated in Figure 2.3. Let d be the extension of the model and f be the generated force along its extension. Assuming that the elastic and the viscous elements are linear, we have

$$f = kd + b\dot{d} \quad (2.1)$$

where constants k and b are denote the elastic and viscous coefficients.

Let us formulate the motion of four particles P_0 through P_3 on a line connected by three deformation elements E_0 through E_2 illustrated in Figure 2.4. Particle P_0 is fixed to space and an external force $f^{\text{ext}}(t)$ is applied to particle P_3 . Let u_0 through u_3 be the displacements of particles P_0 through P_3 . Then, extensions of elements E_0 through E_2 are given by $d_0 = u_1 - u_0$, $d_1 = u_2 - u_1$, and $d_2 = u_3 - u_2$. Let f_0 through f_2 be forces generated by elements E_0 through E_2 along their extensions. Each element applies its force in the positive direction to its left particle and in the negative direction to its right particle. Thus, motion equations of the particles are described as follows:

$$\begin{aligned} m\ddot{u}_0 &= f_0 & + \lambda(t) \\ m\ddot{u}_1 &= f_1 - f_0 \\ m\ddot{u}_2 &= f_2 - f_1 \\ m\ddot{u}_3 &= -f_2 & + f^{\text{ext}}(t) \end{aligned} \quad (2.2)$$

where $\lambda(t)$ represents the constraint force corresponding to a geometric constraint $u_0 = 0$, which is imposed on particle P_0 . Applying the constraint stabilization method (see Appendix D) to the geometric constraint yields

$$\ddot{u}_0 + 2\omega\dot{u}_0 + \omega^2u_0 = 0 \quad (2.3)$$

where ω denotes a predetermined angular frequency. Solving the above five differential equations, we obtain five functions $u_0(t)$ through $u_3(t)$ and $\lambda(t)$.

Assuming that E_0 through E_2 are formulated as a Voigt element with elastic coefficient k and viscous coefficient b , forces f_0 through f_2 are described

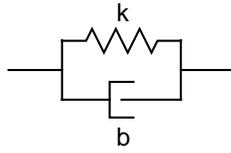


Figure 2.3: Voigt model

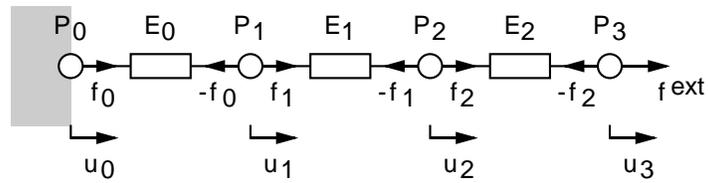
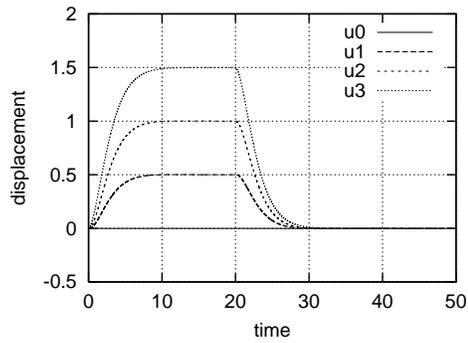
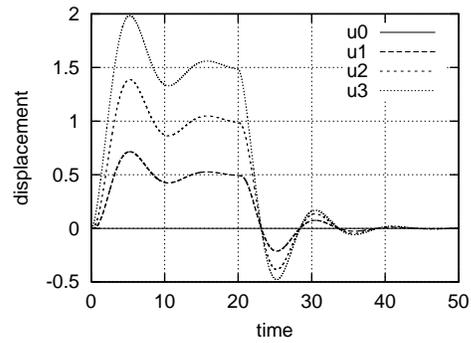


Figure 2.4: Four particles connected by three elements



(a) $m = 1, b = 6, k = 2$



(b) $m = 1, b = 2, k = 2$

Figure 2.5: Motion of four particles connected by Voigt models

as follows:

$$\begin{aligned}
 f_0 &= kd_0 + b\dot{d}_0 = k(u_1 - u_0) + b(\dot{u}_1 - \dot{u}_0), \\
 f_1 &= kd_1 + b\dot{d}_1 = k(u_2 - u_1) + b(\dot{u}_2 - \dot{u}_1), \\
 f_2 &= kd_2 + b\dot{d}_2 = k(u_3 - u_2) + b(\dot{u}_3 - \dot{u}_2).
 \end{aligned}
 \tag{2.4}$$

Solving a set of differential equations (2.2) and (2.3) under a set of forces given in (2.4), we obtain five functions $u_0(t)$ through $u_3(t)$ and $\lambda(t)$. Figure 2.5 shows two examples of the computation. The external force $f^{\text{ext}}(t)$ is equal to 1 at $t \in [0, 20]$ while vanishes thereafter. Parameters are $m = 1$, $b = 6$, and $k = 2$ in Figure 2.5-(a) while $m = 1$, $b = 2$, and $k = 2$ in Figure 2.5-(b). Displacement of each particle converges to a constant as long as a constant external force is applied. This displacement is determined by the elastic coefficient and the magnitude of the external force. For example, during time period $[0, 20]$, displacement u_1 converges to 0.5, which is equal to the magnitude of force divided by the elastic constant 2. Recall that the equivalent elastic coefficient between P_0 and P_2 is given by $2/2$, yielding that displacement u_2 converges to 1, which is equal to the magnitude of force divided by the equivalent elastic constant. Since the equivalent elastic coefficient between P_0 and P_3 is given by $2/3$, displacement u_3 converges to 1.5 during the time period.

2.1.4 Maxwell model

Maxwell model describes viscoplastic deformation of an object. Maxwell model consists of an elastic element and a viscous element connected in serial, as illustrated in Figure 2.6. Let d be the extension of the model and f be the generated force along its extension. Let d^{ela} and d^{vis} be extensions of the elastic and the viscous elements. Assuming that the elastic and the viscous elements are linear, we have

$$\begin{aligned}
 d &= d^{\text{ela}} + d^{\text{vis}}, \\
 f &= kd^{\text{ela}}, \\
 f &= b\dot{d}^{\text{vis}}
 \end{aligned}
 \tag{2.5}$$

where constants k and b denote the elastic and the viscous coefficients. Eliminating d^{vis} yields

$$f = kd^{\text{ela}}, \tag{2.6}$$

$$\dot{d}^{\text{ela}} = -Ad^{\text{ela}} + \dot{d} \tag{2.7}$$

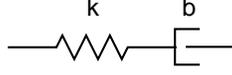


Figure 2.6: Maxwell model

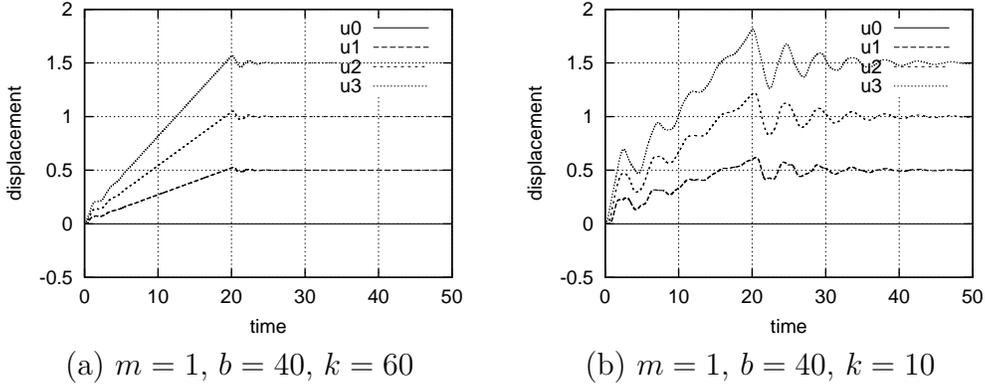


Figure 2.7: Motion of four particles connected by Maxwell models

where $A = k/b$ is a positive constant. The former specifies the force generated by the element. The latter provides a differential equation that a system variable d^{ela} must satisfy.

Let us formulate the motion of four particles P_0 through P_3 on a line connected by three deformation elements E_0 through E_2 illustrated in Figure 2.4. Assuming that E_0 through E_2 are formulated as a Maxwell element with elastic coefficient k and viscous coefficient b , forces f_0 through f_2 are described as follows:

$$f_0 = kd_0^{\text{ela}}, \quad f_1 = kd_1^{\text{ela}}, \quad f_2 = kd_2^{\text{ela}}, \quad (2.8)$$

where variables d_0^{ela} through d_2^{ela} satisfy the following differential equations:

$$\begin{aligned} \dot{d}_0^{\text{ela}} &= -Ad_0^{\text{ela}} + \dot{d}_0 = -Ad_0^{\text{ela}} + (\dot{u}_1 - \dot{u}_0), \\ \dot{d}_1^{\text{ela}} &= -Ad_1^{\text{ela}} + \dot{d}_1 = -Ad_1^{\text{ela}} + (\dot{u}_2 - \dot{u}_1), \\ \dot{d}_2^{\text{ela}} &= -Ad_2^{\text{ela}} + \dot{d}_2 = -Ad_2^{\text{ela}} + (\dot{u}_3 - \dot{u}_2). \end{aligned} \quad (2.9)$$

Solving a set of differential equations (2.2), (2.3), and (2.9) under a set of forces given in (2.8), we obtain eight functions $u_0(t)$ through $u_3(t)$, $d_0^{\text{ela}}(t)$ through $d_2^{\text{ela}}(t)$, and $\lambda(t)$. Figure 2.7 shows two examples of the computation.

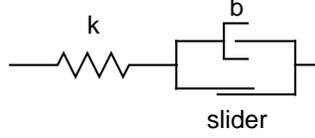


Figure 2.8: Maxwell model with slider

The external force $f^{\text{ext}}(t)$ is equal to 1 at $t \in [0, 20]$ while vanishes thereafter. Parameters are $m = 1$, $b = 40$, and $k = 60$ in Figure 2.7-(a) while $m = 1$, $b = 40$, and $k = 10$ in Figure 2.7-(b). The rate of displacement of each particle converges to a constant as long as a constant external force is applied. This rate is determined by the viscous coefficient and the magnitude of the external force. For example, during time period $[0, 20]$, rate \dot{u}_1 converges to 0.025, which is equal to the magnitude of force divided by the viscous constant 40. Recall that the equivalent viscous coefficient between P_0 and P_2 is given by $40/2$, yielding that rate \dot{u}_2 converges to 0.050, which is equal to the magnitude of force divided by the equivalent viscous constant. Since the equivalent viscous coefficient between P_0 and P_3 is given by $40/3$, rate \dot{u}_3 converges to 0.075 during the time period.

Maxwell model with slider The viscous element of a Maxwell model can extend until an applied force is removed. To avoid such limitless extension, we attach a slider to the viscous element, as illustrated in Figure 2.8. A slider allows the extension of the viscous element while a condition is satisfied. The slider prohibits the extension once the condition is broken. Generally, the third equation of eq.(2.5), which characterizes the motion of the viscous element, is replaced by the following equation:

$$bd^{\text{vis}} = \begin{cases} f & \text{if a condition is satisfied} \\ 0 & \text{otherwise} \end{cases} \quad (2.10)$$

Eliminating d^{vis} yields

$$\text{if a condition is satisfied} \quad \begin{cases} f = kd^{\text{ela}}, \\ \dot{d}^{\text{ela}} = \dot{d} - Ad^{\text{ela}}. \end{cases} \quad (2.11)$$

$$\text{otherwise} \quad \begin{cases} f = kd^{\text{ela}}, \\ \dot{d}^{\text{ela}} = \dot{d}. \end{cases} \quad (2.12)$$

Note that $b \rightarrow \infty$ yields $A \rightarrow 0$, converting eq.(2.11) into eq.(2.12).

Let l be the length of a Maxwell model with a slider and l^{ela} be the length of its elastic element. A simple condition for the slider restricts the length of the elastic element:

$$0 \leq l^{\text{ela}} \leq l.$$

Generally,

$$a_{\min}l \leq l^{\text{ela}} \leq a_{\max}l$$

where a_{\min} and a_{\max} are constants not less than 0 and not more than 1. These parameters define the possible minimum and maximum values of ratio l^{ela}/l . Let l_{init} be the initial length of the element and $l_{\text{init}}^{\text{ela}}$ be the initial length of its elastic element. The above condition can be rewritten as follows:

$$a_{\min}(l_{\text{init}} + d) \leq l_{\text{init}}^{\text{ela}} + d^{\text{ela}} \leq a_{\max}(l_{\text{init}} + d).$$

A viscous element with a slider stops extending when it reaches length thresholds determined by a_{\min} and a_{\max} , even though an external force may still be there.

2.1.5 Three-element model

Three-element model describes viscoelastic and viscoplastic deformation of an object, which is referred to as *rheological deformation*. Three-element model consists of a Voigt element and a viscous element connected in serial, as illustrated in Figure 2.9. Let d be the extension of the model and f be the generated force along its extension. Let d^{voigt} and d^{vis} be extensions of the Voigt and the viscous elements. Assuming that the elastic and the viscous elements are linear, we have

$$\begin{aligned} d &= d^{\text{voigt}} + d^{\text{vis}}, \\ f &= kd^{\text{voigt}} + bd\dot{d}^{\text{voigt}}, \\ f &= b'\dot{d}^{\text{vis}} \end{aligned} \tag{2.13}$$

where constant k represents an elastic coefficient and constants b and b' denote viscous coefficients. Eliminating d^{vis} yields

$$f = B(kd^{\text{voigt}} + b\dot{d}), \tag{2.14}$$

$$\dot{d}^{\text{voigt}} = -Ad^{\text{voigt}} + B\dot{d} \tag{2.15}$$

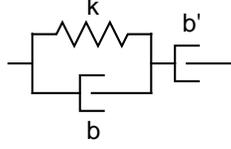


Figure 2.9: Three-element model

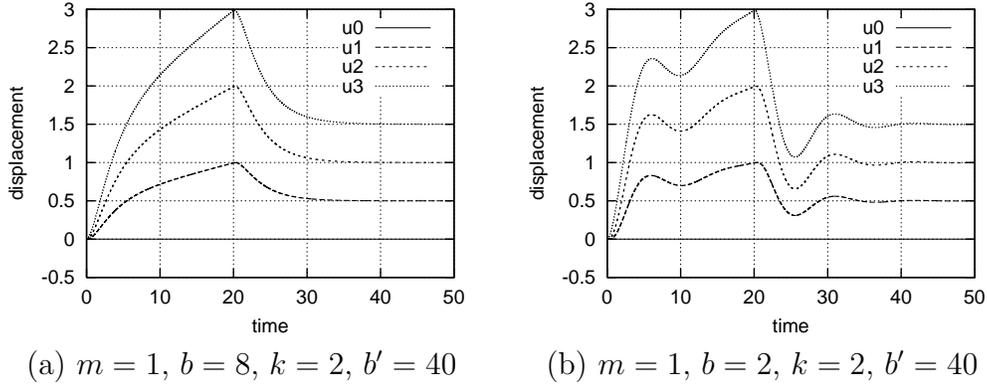


Figure 2.10: Motion of four particles connected by three-element models

where $A = k/(b + b')$ and $B = b/(b + b')$ are positive constants. The former specifies the force generated by the element. The latter provides a differential equation that a system variable d^{voigt} must satisfy.

Let us formulate the motion of four particles P_0 through P_3 on a line connected by three deformation elements E_0 through E_2 illustrated in Figure 2.4. Assuming that E_0 through E_2 are formulated as a three element model with elastic coefficient k and viscous coefficients b and b' , forces f_0 through f_2 are described as follows:

$$\begin{aligned}
 f_0 &= B \{ k d_0^{\text{voigt}} + b \dot{d}_0 \} = B \{ k d_0^{\text{voigt}} + b(\dot{u}_1 - \dot{u}_0) \}, \\
 f_1 &= B \{ k d_1^{\text{voigt}} + b \dot{d}_1 \} = B \{ k d_1^{\text{voigt}} + b(\dot{u}_2 - \dot{u}_1) \}, \\
 f_2 &= B \{ k d_2^{\text{voigt}} + b \dot{d}_2 \} = B \{ k d_2^{\text{voigt}} + b(\dot{u}_3 - \dot{u}_2) \},
 \end{aligned} \tag{2.16}$$

where variables d_0^{voigt} through d_2^{voigt} satisfy the following differential equations:

$$\dot{d}_0^{\text{voigt}} = -A d_0^{\text{voigt}} + B \dot{d}_0 = -A d_0^{\text{voigt}} + B(\dot{u}_1 - \dot{u}_0),$$

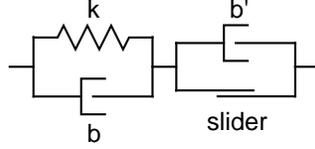


Figure 2.11: Three-element model with slider

$$\begin{aligned} \dot{d}_1^{\text{voigt}} &= -Ad_1^{\text{voigt}} + B\dot{d}_1 = -Ad_1^{\text{voigt}} + B(\dot{u}_2 - \dot{u}_1), \\ \dot{d}_2^{\text{voigt}} &= -Ad_2^{\text{voigt}} + B\dot{d}_2 = -Ad_2^{\text{voigt}} + B(\dot{u}_3 - \dot{u}_2). \end{aligned} \quad (2.17)$$

Solving a set of differential equations (2.2), (2.3), and (2.17) under a set of forces given in (2.16), we obtain eight functions $u_0(t)$ through $u_3(t)$, $d_0^{\text{voigt}}(t)$ through $d_2^{\text{voigt}}(t)$, and $\lambda(t)$. Figure 2.10 shows two examples of the computation. The external force $f^{\text{ext}}(t)$ is equal to 1 at $t \in [0, 20]$ while vanishes thereafter. Parameters are $m = 1$, $b = 8$, $k = 2$, and $b' = 40$ in Figure 2.10-(a) while $m = 1$, $b = 2$, $k = 2$, and $b' = 40$ in Figure 2.10-(b). Note that a three-element model consists of the Voigt part, which is characterized by k and b , and the viscous part, which is specified by b' . Both parts deform while an external force is applied to the model. After the external force is released, deformation of the Voigt part vanishes but the viscous part contains its deformation. During time period $[0, 20]$, the extension of each Voigt part converges to 0.5, which is equal to f^{ext}/k . Extension rate of each viscous part converges to $1/40$, which is equal to f^{ext}/b' , implying that the extension of the viscous part reaches to 0.5 during the time period. Consequently, the extension of each three-element model is equal to the sum of the two extensions, that is, 1.0 at time $t = 20$. Extension of each Voigt part is lost after an external force is released at $t = 20$ but the extension of each viscous part remains. As a result, the extension of each three-element model converges to 0.5 after an external force is released.

Three-element model with slider The viscous element of a three-element model can extend until an applied force is removed. To avoid such limitless extension, we use a three-element model with a slider illustrated in Figure 2.11. A slider allows the extension of the viscous element while a condition is satisfied. The slider prohibits the extension once the condition is broken. Generally, the third equation of eq.(2.13), which characterizes the motion of

the viscous element, is replaced by the following equation:

$$b'd^{\text{vis}} = \begin{cases} f & \text{if a condition is satisfied} \\ 0 & \text{otherwise} \end{cases} \quad (2.18)$$

Eliminating d^{vis} yields

$$\text{if a condition is satisfied} \quad \begin{cases} f = B(kd^{\text{voigt}} + b\dot{d}), \\ \dot{d}^{\text{voigt}} = -Ad^{\text{voigt}} + B\dot{d}. \end{cases} \quad (2.19)$$

$$\text{otherwise} \quad \begin{cases} f = kd^{\text{voigt}} + b\dot{d}, \\ \dot{d}^{\text{voigt}} = \dot{d}. \end{cases} \quad (2.20)$$

Note that $b' \rightarrow \infty$ yields $A \rightarrow 0$ and $B \rightarrow 1$, converting eq.(2.19) into eq.(2.20).

Let l be the length of a three-element model with a slider and l^{voigt} be the length of its Voigt part. A simple condition for the slider restricts the length of the Voigt part:

$$0 \leq l^{\text{voigt}} \leq l.$$

Generally,

$$a_{\min}l \leq l^{\text{voigt}} \leq a_{\max}l$$

where a_{\min} and a_{\max} are constants not less than 0 and not more than 1. These parameters define the possible minimum and maximum values of ratio l^{voigt}/l . Let l_{init} be the initial length of the element and $l_{\text{init}}^{\text{voigt}}$ be the initial length of its Voigt part. The above condition can be rewritten as follows:

$$a_{\min}(l_{\text{init}} + d) \leq l_{\text{init}}^{\text{voigt}} + d^{\text{voigt}} \leq a_{\max}(l_{\text{init}} + d).$$

A viscous element with a slider stops extending when it reaches length thresholds determined by a_{\min} and a_{\max} , even though an external force may still be there.

2.1.6 Generalized Voigt model

Applying an external force to a deformable object and releasing the force yield the *free deformation* of the object. Free deformation of a single Voigt

model is basically determined by a single value k/b . Unfortunately, free deformation of an actual object often shows different behavior, which cannot be characterized by a single value of k/b . This behavior can be described by a set of Voigt models connected in serial, which is referred to as *generalized Voigt model*.

Let us formulate the generalized Voigt model illustrated in Figure 2.12. This model consists of n Voigt elements. Let d be the extension of the generalized Voigt model while d_i be the extension of the i -th element. Extensions d and d_1 through d_{n-1} are independent state variables. Extension d_n is dependently described by the state variables as follows:

$$d_n = d - d_1 - \cdots - d_{n-1}.$$

Let k_i and b_i be the elastic and viscous coefficients of the i -th Voigt element. Noting that forces generated by individual elements are equal with each other, we have

$$\begin{aligned} f &= k_1 d_1 + b_1 \dot{d}_1, \\ f &= k_2 d_2 + b_2 \dot{d}_2, \\ &\vdots \\ f &= k_n d_n + b_n \dot{d}_n \end{aligned}$$

where f denotes the force generated by the generalized Voigt model. Dividing the above equations by b_1 through b_n and summing up all equations yields

$$f = \frac{\left(\frac{k_1}{b_1} d_1 + \cdots + \frac{k_{n-1}}{b_{n-1}} d_{n-1} \right) + \frac{k_n}{b_n} (d - d_1 - \cdots - d_{n-1}) + \dot{d}}{\frac{1}{b_1} + \frac{1}{b_2} + \cdots + \frac{1}{b_n}}.$$

Note that using the above equation, force f can be computed from state variables d_1 through d_{n-1} and \dot{d} . This equation specifies the force generated by a generalized Voigt model. State variables d_1 through d_{n-1} satisfy the following differential equations

$$\begin{aligned} \dot{d}_1 &= -\frac{k_1}{b_1} d_1 + \frac{1}{b_1} f, \\ \dot{d}_2 &= -\frac{k_2}{b_2} d_2 + \frac{1}{b_2} f, \end{aligned}$$

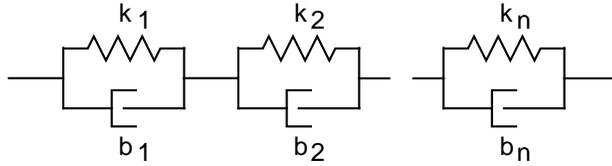


Figure 2.12: Generalized Voigt model

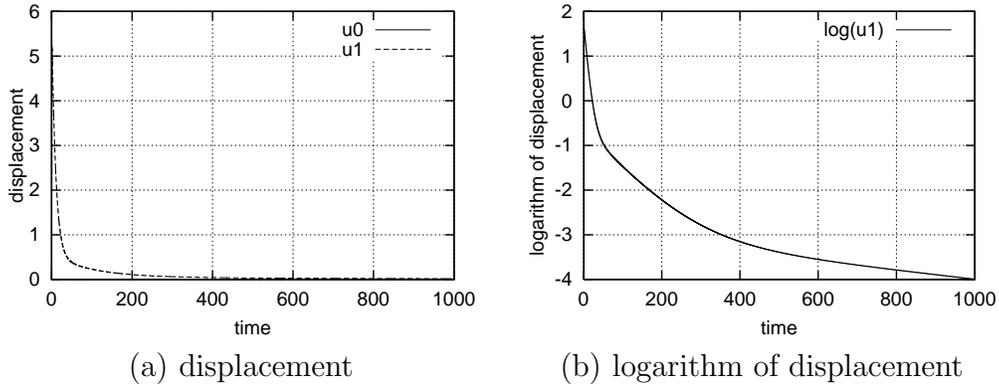


Figure 2.13: Free deformation of generalized Voigt model

$$\dot{d}_{n-1} = -\frac{k_{n-1}}{b_{n-1}}d_{n-1} + \frac{1}{b_{n-1}}f.$$

Consequently, we can construct dynamic equations of particles connected by generalized Voigt models.

Let us simulate the free deformation of a generalized Voigt model connecting a particle and a fixed wall. The model consists of three Voigt elements; of which elastic and viscous coefficients are given by $k_1/b_1 = 1/10$, $k_2/b_2 = 1/100$, and $k_3/b_3 = 1/1000$. Mass of the particle is equal to 1. External force applied to the particle is equal to 50 at $t \in [0, 1]$ while vanishes thereafter. Displacement of the particle is plotted in Figure 2.13-(a). Logarithm of the displacement is also plotted in Figure 2.13-(b). As plotted in the figure, logarithm of the displacement can be approximated by three line segments; each corresponds to one of three Voigt element.

2.1.7 Generalized Maxwell model

While a constant deformation is exerted to an object, force generated by the deformed object decreases. This phenomena is referred to as *force relaxation*. Force relaxation of a single Maxwell model is basically determined by a single value k/b . Unfortunately, force relaxation of an actual object often shows different property, which cannot be characterized by a single value of k/b . This property can be described by a set of Maxwell models connected in parallel, which is referred to as *generalized Maxwell model*.

Let us formulate the generalized Maxwell model illustrated in Figure 2.14. This model consists of n Maxwell elements. Let f be the force generated by the generalized Voigt model and f_i be the force generated by the i -th element. Let k_i and b_i be the elastic and viscous coefficients of the i -th Maxwell element. Let d_1^{ela} through d_n^{ela} be extensions of elastic elements specified by k_1 through k_n . Then, we have

$$f = k_1 d_1^{\text{ela}} + k_2 d_2^{\text{ela}} + \cdots + k_n d_n^{\text{ela}}.$$

Note that using the above equation, force f can be computed from state variables d_1^{ela} through d_n^{ela} . This equation specifies the force generated by a generalized Maxwell model. Noting that the rate of extension in individual elements are equal with each other, we have

$$\begin{aligned} \dot{d}_1^{\text{ela}} &= -\frac{k_1}{b_1} d_1^{\text{ela}} + \dot{d}, \\ \dot{d}_2^{\text{ela}} &= -\frac{k_2}{b_2} d_2^{\text{ela}} + \dot{d}, \\ &\vdots \\ \dot{d}_n^{\text{ela}} &= -\frac{k_n}{b_n} d_n^{\text{ela}} + \dot{d} \end{aligned}$$

where \dot{d} denotes the rate of extension of the generalized Maxwell model. The above equations provide a set of differential equations that state variables d_1^{ela} through d_n^{ela} must satisfy. Consequently, we can construct dynamic equations of particles connected by generalized Maxwell models.

Let us simulate the force relaxation of a generalized Maxwell model connecting a particle and a fixed wall. The model consists of three Maxwell elements; of which elastic and viscous coefficients are given by $k_1/b_1 = 1/10$, $k_2/b_2 = 1/100$, and $k_3/b_3 = 1/1000$. Mass of the particle is equal to 1. The

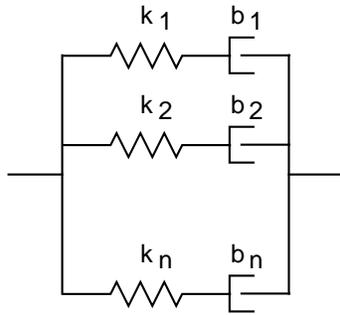


Figure 2.14: Generalized Maxwell model

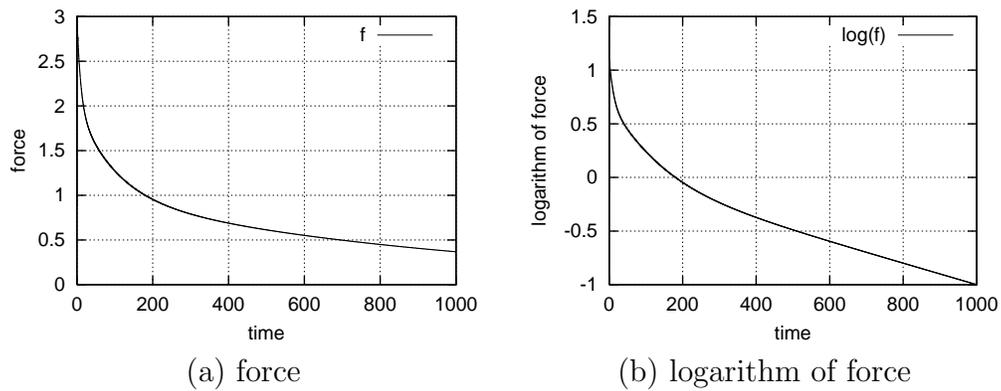


Figure 2.15: Force relaxation of generalized Maxwell model

particle is forced to move toward the wall by displacement 1. Force applied to the particle is plotted in Figure 2.15-(a). Logarithm of the force is also plotted in Figure 2.15-(b). As plotted in the figure, logarithm of the force can be approximated by three line segments; each corresponds to one of three Maxwell element.

2.2 Particle-Based Modeling of 2D Object Deformation

2.2.1 Geometric description of 2D object

This section describes geometric description of a 2D object for its particle-based modeling. Any 2D shape can be approximated by a set of triangles. Each triangle consists of three particles and three edges. Each edge connects two particles; one is referred to as the starting particle of the edge while the other is referred to as its end particle. This implies that each edge is directed from the starting particle to the end particle. Figure 2.16 shows a simple 2D object. This object consists of 4 particles, 5 edges, and 2 triangles. Table 2.1 provides the geometric description of the object. The starting and end particles are specified for each edge. Let us trace the contour of a triangle counterclockwise. Some edges are in the positive direction during the tracing while the others are in the negative direction. The former is denoted by symbol $+$ and the latter is expressed by symbol $-$ in the table.

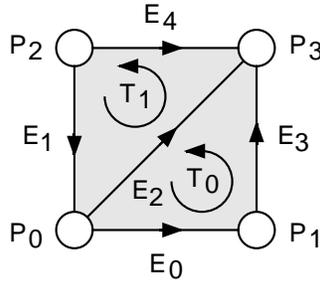


Figure 2.16: Example of 2D object

Let us formulate the extension of each edge and its time rate of change. Let P_i and P_j be the starting and end particles of edge E_a . Let $\mathbf{x}_i(t) = [x_i, y_i]^T$ be the position of particle P_i and $\mathbf{v}_i(t) = [v_{ix}, v_{iy}]$ be its velocity at time t . Let $\mathbf{x}_i^{\text{init}} = [x_i^{\text{init}}, y_i^{\text{init}}]^T$ be the natural position of particle P_i . Note that the natural position $\mathbf{x}_i^{\text{init}}$ is constant. Let l_a denote the current length of edge E_a and l_a^{init} be its natural length, which are described as

$$l_a = \{(\mathbf{x}_i - \mathbf{x}_j) \cdot (\mathbf{x}_i - \mathbf{x}_j)\}^{\frac{1}{2}}, \quad (2.21)$$

$$l_a^{\text{init}} = \{(\mathbf{x}_i^{\text{init}} - \mathbf{x}_j^{\text{init}}) \cdot (\mathbf{x}_i^{\text{init}} - \mathbf{x}_j^{\text{init}})\}^{\frac{1}{2}}. \quad (2.22)$$

Table 2.1: Description of 2D object by a set of triangles

edge	starting particle	end particle
E ₀	P ₀	P ₁
E ₁	P ₂	P ₀
E ₂	P ₀	P ₃
E ₃	P ₁	P ₃
E ₄	P ₂	P ₃

triangle	edges		
T ₀	E ₀ +	E ₃ -	E ₂ -
T ₁	E ₂ +	E ₄ -	E ₁ +

Note that length l_a^{init} is constant. Extension of edge E_a at time t is simply described as follows:

$$d_a = l_a - l_a^{\text{init}}. \quad (2.23)$$

Differentiating the above equations with respect to time yields the time rate of change of extension d_a :

$$\dot{d}_a = (\mathbf{v}_i - \mathbf{v}_j) \cdot \mathbf{e}_a \quad (2.24)$$

where \mathbf{e}_a denotes the unit vector from the starting particle to the end particle, which is given by

$$\mathbf{e}_a = \frac{\mathbf{x}_i - \mathbf{x}_j}{l_a}. \quad (2.25)$$

Let us formulate the signed area of each triangle and its time rate of change. Let P_i , P_j , and P_k be the particles of triangle T_p . The area is positive if the triangular loop is counter clockwise while is negative if the loop is clockwise. The signed area of $\triangle OP_iP_j$ is formulated as

$$\triangle OP_iP_j = \frac{1}{2} \begin{vmatrix} x_i & x_j \\ y_i & y_j \end{vmatrix}.$$

Let S_p be the signed area of triangle T_p , which is given as follows:

$$S_p = \triangle OP_iP_j + \triangle OP_jP_k + \triangle OP_kP_i.$$

Differentiating the above equations with respect to time yields the time rate of change of extension S_p :

$$\dot{S}_p = \frac{d}{dt} \triangle OP_iP_j + \frac{d}{dt} \triangle OP_jP_k + \frac{d}{dt} \triangle OP_kP_i$$

where

$$\frac{d}{dt} \Delta OP_i P_j = \frac{1}{2} \begin{vmatrix} v_{ix} & x_j \\ v_{iy} & y_j \end{vmatrix} + \frac{1}{2} \begin{vmatrix} x_i & v_{jx} \\ y_i & v_{jy} \end{vmatrix}.$$

2.2.2 Truss model

Let us attach deformation elements to individual edges in a 2D object shape. This model is referred to to a *truss model*. We can formulate forces caused by the attached deformation elements and derive dynamic equations of particles, implying that we can sketch the object deformation by solving the dynamic equations. This section describes the 2D truss model of a deformable object.

Let us formulate a set of dynamic equations of particles P_0 through P_3 in a 2D object shown in Figure 2.16. Particle P_0 is fixed to space and an external force $\mathbf{f}^{\text{ext}}(t)$ is applied to particle P_3 . Let m_i be the mass of particle P_i . Deformation elements are attached to individual edges. Let f_a be the magnitude of force generated by the deformation element corresponding to edge E_a along its extension. Then, the deformation element applies force $f_a \mathbf{e}_a$ to the starting particle of the edge and force $-f_a \mathbf{e}_a$ to its end particle. As a result, we have the following dynamic equations of particles:

$$\begin{aligned} m_0 \ddot{\mathbf{u}}_0 &= f_0 \mathbf{e}_0 - f_1 \mathbf{e}_1 + f_2 \mathbf{e}_2 && + \boldsymbol{\lambda}(t) \\ m_1 \ddot{\mathbf{u}}_1 &= -f_0 \mathbf{e}_0 && + f_3 \mathbf{e}_3 \\ m_2 \ddot{\mathbf{u}}_2 &= f_1 \mathbf{e}_1 && + f_4 \mathbf{e}_4 \\ m_3 \ddot{\mathbf{u}}_3 &= -f_2 \mathbf{e}_2 - f_3 \mathbf{e}_3 - f_4 \mathbf{e}_4 + \mathbf{f}^{\text{ext}}(t) \end{aligned} \quad (2.26)$$

where $\boldsymbol{\lambda}(t)$ represents the constraint force corresponding to a geometric constraint $\mathbf{u}_0 = \mathbf{0}$, which is imposed on particle P_0 . Applying the constraint stabilization method (see Appendix D) to the geometric constraint yields

$$\ddot{\mathbf{u}}_0 + 2\omega \dot{\mathbf{u}}_0 + \omega^2 \mathbf{u}_0 = \mathbf{0} \quad (2.27)$$

where ω denotes a predetermined angular frequency. Solving the above five differential equations, we obtain five functions $\mathbf{u}_0(t)$ through $\mathbf{u}_3(t)$ and $\boldsymbol{\lambda}(t)$.

Voigt model

Let us attach a Voigt model specified by elastic coefficient k and viscous coefficient b to all edges. Force f_a is then simply described as $f_a = kd_a + b\dot{d}_a$. Substituting $f_0 = kd_0 + b\dot{d}_0$ through $f_4 = kd_4 + b\dot{d}_4$ into eq.(2.26), we have

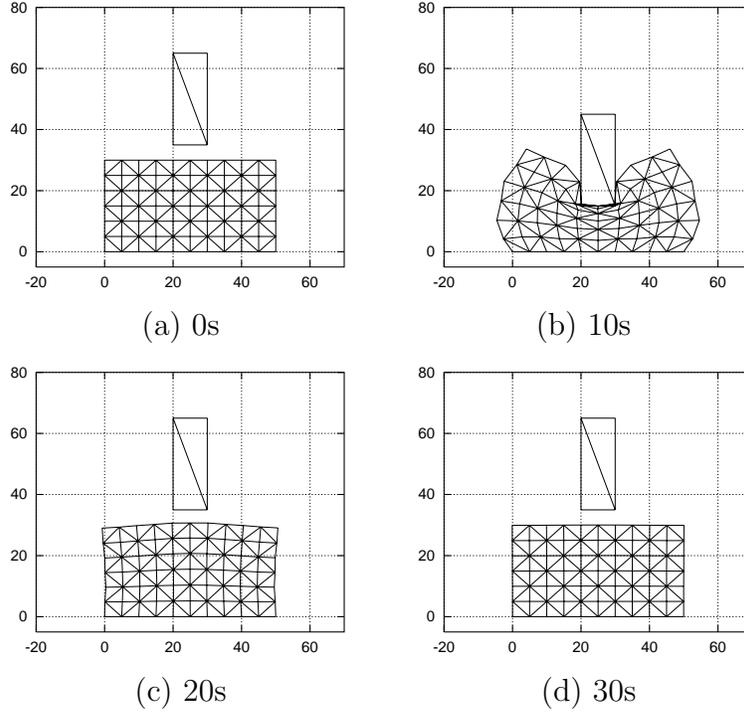


Figure 2.17: Computation of 2D Voigt deformation

four differential equations of the second order with respect to \mathbf{u}_0 through \mathbf{u}_3 . Solving the four differential equations with eq.(2.27), we can obtain \mathbf{u}_0 through \mathbf{u}_3 and $\boldsymbol{\lambda}$ to sketch the 2D Voigt deformation.

Figure 2.17 demonstrates 2D Voigt deformation. An elastic object on a flat table is deformed by a rigid bar. The rigid bar moves downward during the first 10s and moves upward during the next 10s. The elastic object is described by a set of 16×6 squares, each of which is divided into two triangles. Elastic and viscous coefficients of the Voigt model are given by $k = 1.2$ and $b = 1.5$. Total mass 30 is equally distributed to all 17×7 particles. All bottom particles of the elastic object are fixed to the table. Topology maintenance forces (see Section 2.2.4) and volumatic forces (see Section 2.2.5) are introduced to this simulation. As illustrated in the figure, the simulation describes elastic nature of the object.

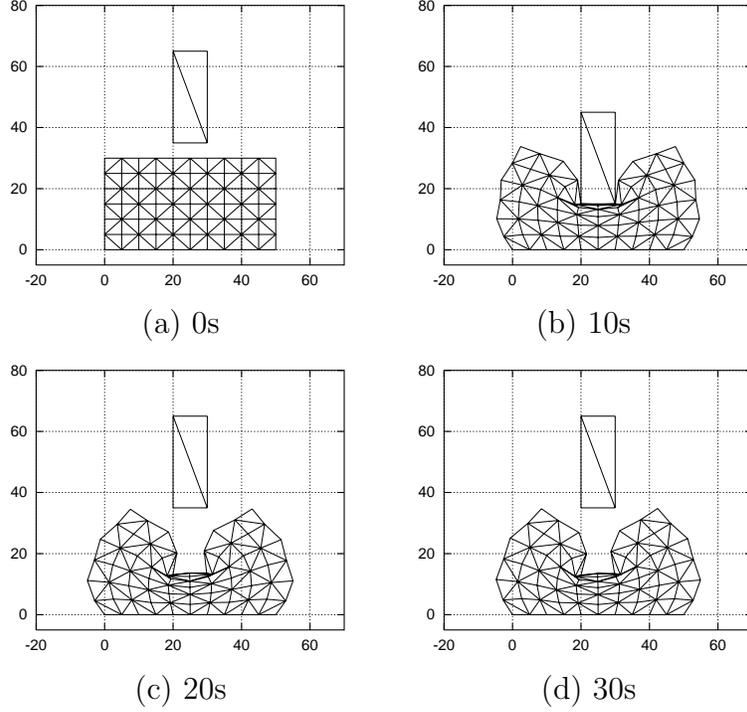


Figure 2.18: Computation of 2D Maxwell deformation

Maxwell model

Let us attach a Maxwell model specified by elastic coefficient k and viscous coefficient b to all edges. Let d_a^{ela} be the extension of the elastic element of the Maxwell model on the a -th edge. Force f_a is then described as $f_a = kd_a^{\text{ela}}$ where $A = k/b$. Substituting $f_0 = kd_0^{\text{ela}}$ through $f_4 = kd_4^{\text{ela}}$ into eq.(2.26), we have four differential equations of the second order with respect to \mathbf{u}_0 through \mathbf{u}_3 . In addition, extension d_a^{ela} satisfies

$$\dot{d}_a^{\text{ela}} = -Ad_a^{\text{ela}} + \dot{d}_a.$$

Thus, we have five differential equations of the first order with respect to d_0^{ela} through d_4^{ela} . Solving the four differential equations of the second order and the five differential equations of the first order with eq.(2.27), we can obtain \mathbf{u}_0 through \mathbf{u}_3 , d_0^{ela} through d_4^{ela} , and λ to sketch the 2D Maxwell deformation.

Figure 2.18 demonstrates 2D Maxwell deformation. Elastic and viscous coefficients of the Maxwell model are given by $k = 1.2$ and $b = 1.5$. Total

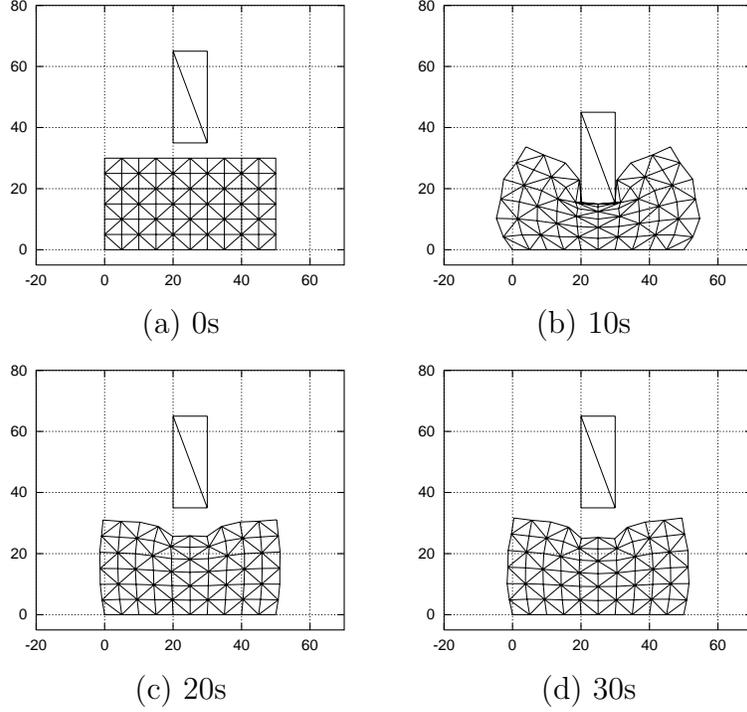


Figure 2.19: Computation of 2D three-element deformation

mass 30 is equally distributed to all particles. As illustrated in the figure, the simulation describes viscoplastic nature of the object.

Three-element model

Let us attach a three-element model specified by elastic coefficient k and viscous coefficients b and b' to all edges. Let d_a^{ela} be the extension of the elastic element of the Maxwell model on the a -th edge. Force f_a is then described as $f_a = B(kd_a^{\text{voigt}} + bd_a)$, where $A = k/(b + b')$ and $B = b'/(b + b')$. Substituting $f_0 = B(kd_0^{\text{voigt}} + bd_0)$ through $f_4 = B(kd_4^{\text{voigt}} + bd_4)$ into eq.(2.26), we have four differential equations of the second order. In addition, extension d_a^{voigt} satisfies

$$\dot{d}_a^{\text{voigt}} = -Ad_a^{\text{voigt}} + Bd_a.$$

Thus, we have five differential equations of the first order with respect to d_0^{voigt} through d_4^{voigt} . Solving the four differential equations of the second order and

the five differential equations of the first order with eq.(2.27), we can obtain \mathbf{u}_0 through \mathbf{u}_3 , d_0^{voigt} through d_4^{voigt} , and $\boldsymbol{\lambda}$ to sketch the 2D three-element deformation.

Figure 2.19 demonstrates 2D three-element deformation. Elastic and viscous coefficients of the Maxwell model are given by $k = 1.2$, $b = 1.5$, and $b' = 15$. Total mass 30 is equally distributed to all particles. As illustrated in the figure, the simulation describes rheological nature of the object.

2.2.3 Internal forces among particles

Recall that a system of particles should conserve its linear momentum and angular momentum as long as no external forces are applied to the system. In other words, the sum of all internal forces and the sum of moments of all internal forces must vanish. Consequently, we must satisfy these two conditions in introducing internal forces to a particle-based model. Deformation element attached to each individual edge applies a pair of forces to its starting and end particles. These two forces, which are referred to as *edge forces*, are internal.

Let us introduce a pair of forces applied to a particle and its opposite edge. Let P_i be a particle and E_a be its opposite edge, of which starting and end particles are denoted by P_j and P_k . Let $\mathbf{f}_{i,a}$ be a force applied to particle P_i by edge E_a . Then, its negative force $-\mathbf{f}_{i,a}$ should be applied to edge E_a at point P, where point P is the intersection point of a line including edge E_a and another line that passes point P_i along directional vector $\mathbf{f}_{i,a}$. Force $-\mathbf{f}_{i,a}$ applied to edge E_a at point P can be equivalently split into two forces applied to particles P_j and P_k . Consequently, the following three forces, which are referred to as *particle-edge forces*, are applied to particles P_i , P_j , and P_k :

$$\begin{aligned}\mathbf{f}_{i,a; i}^{\text{pe}} &= \mathbf{f}_{i,a}, \\ \mathbf{f}_{i,a; j}^{\text{pe}} &= \frac{\beta}{\alpha + \beta}(-\mathbf{f}_{i,a}), \\ \mathbf{f}_{i,a; k}^{\text{pe}} &= \frac{\alpha}{\alpha + \beta}(-\mathbf{f}_{i,a})\end{aligned}$$

where α and β are signed distances given by $\alpha = \overrightarrow{P_j P} \cdot \mathbf{e}_a$ and $\beta = \overrightarrow{P P_k} \cdot \mathbf{e}_a$. Since the sum of the three forces and the sum of their moments vanish, particle-edge forces are internal as well.

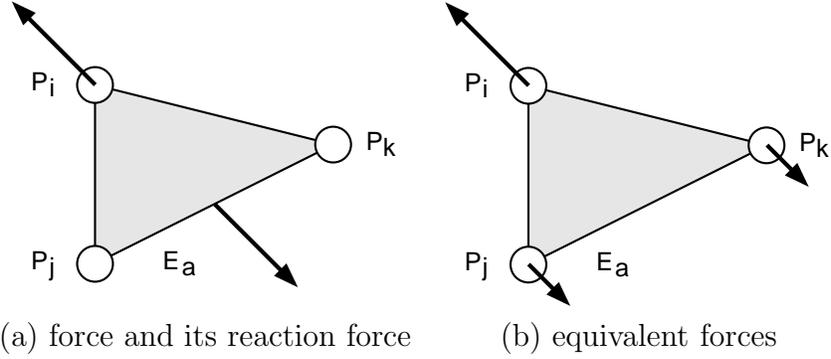


Figure 2.20: Particle-edge force

2.2.4 Topology maintenance force

Since the dynamic equations are solved numerically at discrete times, a triangle in the particle-based model may reverse, resulting in failure of the computation. Let us define a particle-edge force that avoids the reverse of triangles in the model. Let us introduce vector \mathbf{n}_a at edge E_a so that \mathbf{e}_a and \mathbf{n}_a form a right-handed coordinate system. Let T_p be a triangle consisting of particle P_i and its opposite edge E_a . Let P_j and P_k be the starting and end particles of the edge.

Assume that edge E_a is in positive direction along the triangle. Vector \mathbf{n}_a is the inward normal vector of edge E_a at the initial shape of an object, as illustrated in Figure 2.21-(a-1). Thus, it is still the inward normal vector while triangle T_p be consistent. Contrary, vector \mathbf{n}_a turns to the outward normal vector when triangle T_p reverses, as shown in Figure 2.21-(a-2). Let us introduce a signed distance between particle P_i and edge E_a given by

$$d_{i,a} = \overrightarrow{P_j P_i} \cdot \mathbf{n}_a. \quad (2.28)$$

The signed distance is positive while triangle T_p is consistent but is negative when the triangle reverses. In other words, when the signed distance falls negative, particle P_i must be guided so that the signed distance increases. Thus, we can introduce the following artificial force generated by a virtual Voigt model between particle P_i and edge E_a to avoid the reverse of the triangle:

$$\mathbf{f}_{i,a} = \begin{cases} \mathbf{0} & (d_{i,a} \geq 0) \\ \{-Kd_{i,a} - C\dot{d}_{i,a}\}\mathbf{n}_a & (d_{i,a} < 0) \end{cases} \quad (2.29)$$

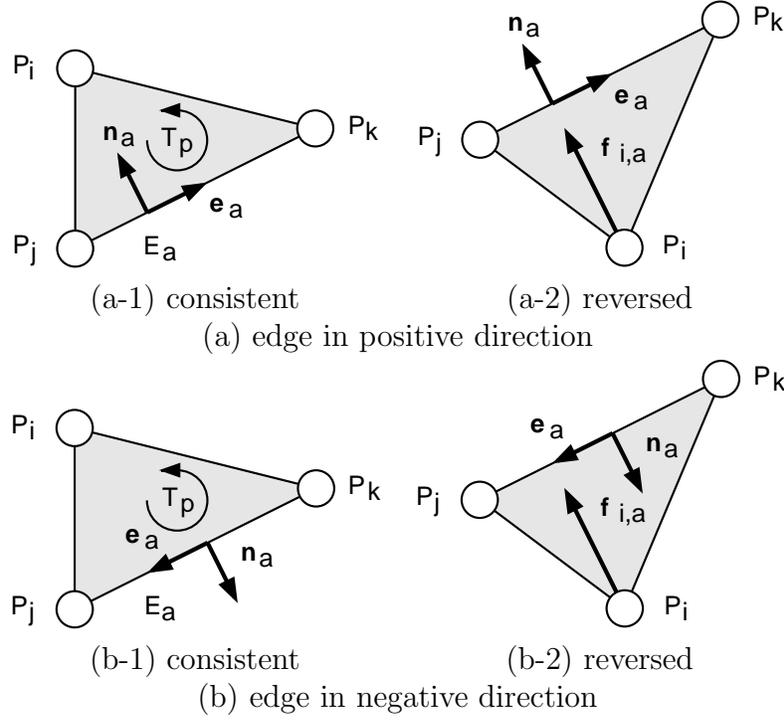


Figure 2.21: Signed distance between particle and its opposite edge

where K and C denote elastic and viscous coefficients. When the signed distance $d_{i,a}$ drops negative, the above artificial force and its reaction force are applied to particle P_i and edge E_a to increase the signed distance.

Assume that edge E_a is in negative direction along triangle T_p , as illustrated in Figure 2.21-(b-1). Vector \mathbf{n}_a is the outward normal vector while triangle T_p be consistent. Contrary, vector \mathbf{n}_a turns to the inward normal vector when triangle T_p reverses, as shown in Figure 2.21-(b-2). The signed distance between particle P_i and edge E_a given by

$$d_{i,a} = \overrightarrow{P_j P_i} \cdot (-\mathbf{n}_a) \quad (2.30)$$

is positive while triangle T_p is consistent but is negative when the triangle reverses. Thus, the following force avoids the reverse of the triangle:

$$\mathbf{f}_{i,a} = \begin{cases} \mathbf{0} & (d_{i,a} \geq 0) \\ \{-Kd_{i,a} - Cd_{i,a}\}(-\mathbf{n}_a) & (d_{i,a} < 0) \end{cases} \cdot \quad (2.31)$$

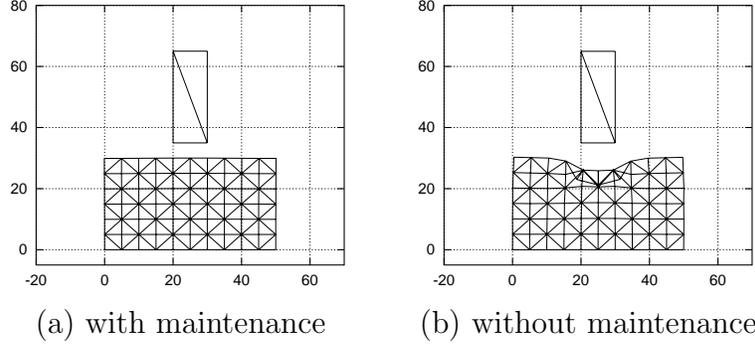


Figure 2.22: Elastic deformation with/without topology maintenance forces

When the signed distance $d_{i,a}$ drops negative, the above artificial force and its reaction force are applied to particle P_i and edge E_a to increase the signed distance.

Consequently, a set of particle-edge forces caused by force $\mathbf{f}_{i,a}$ defined in either eq.(2.29) or (2.31) prevents the reverse of triangles.

Figures 2.22-(a) and (b) demonstrate how topology maintenance forces work. An elastic triangular object with $k = 1.2$ and $b = 1.5$ is deformed by a rigid bar. Each figure shows the stationary shape of the elastic object after its deformation. Topology maintenance forces are introduced in the computation of Figure 2.22-(a) but not in Figure 2.22-(b). As shown in 2.22-(a), the stationary shape coincides to the natural shape of the object. On the other hand, shape shown in 2.22-(b) collapses, that is, triangles near the contacting region reverse. This demonstrates that the topology maintenance forces are requisite to prevent the reverse of triangles in a truss model.

2.2.5 Volumatic force

Let P be pressure inside triangle T_p , which consists of particles P_i , P_j , and P_k . Let E_a , E_b , and E_c are opposite edges of the three particles. Pressure P generates forces acting on the edges of the triangle. The force is applied at the mid-point of an edge along its normal. The magnitude of the force is proportional to the length of the edge. Thus, force applied to edge E_a is given by $Pl_a\mathbf{n}_a^{\text{out}}$, where $\mathbf{n}_a^{\text{out}}$ denotes the outward unit normal vector of edge

E_a :

$$\mathbf{n}_a^{\text{out}} = \begin{cases} \mathbf{n}_a & \text{edge } E_a \text{ is in positive direction} \\ -\mathbf{n}_a & \text{edge } E_a \text{ is in negative direction} \end{cases} .$$

This force can be equivalently split into two forces applied to particles P_j and P_k ; both forces are described as $(1/2)Pl_a\mathbf{n}_i^{\text{out}}$. Forces applied to edges E_b and E_c are also described by $Pl_b\mathbf{n}_b^{\text{out}}$ and $Pl_c\mathbf{n}_c^{\text{out}}$, which can be distributed to their starting and end particles. Consequently, the following forces, which are referred to as *areal forces*, are applied to particles P_i , P_j , and P_k :

$$\begin{aligned} \mathbf{f}_{p; i}^{\text{vol}} &= \frac{1}{2}P \{l_b\mathbf{n}_b^{\text{out}} + l_c\mathbf{n}_c^{\text{out}}\}, \\ \mathbf{f}_{p; j}^{\text{vol}} &= \frac{1}{2}P \{l_c\mathbf{n}_c^{\text{out}} + l_a\mathbf{n}_a^{\text{out}}\}, \\ \mathbf{f}_{p; k}^{\text{vol}} &= \frac{1}{2}P \{l_a\mathbf{n}_a^{\text{out}} + l_b\mathbf{n}_b^{\text{out}}\}. \end{aligned}$$

Note that the above three forces pass the center of circumcircle of triangle T_p . Since the sum of the three forces and the sum of their moments vanish, areal forces are internal.

Formulating pressure P characterizes the object deformation. Let S be the current area of a triangle and S^{init} be its initial area. Voigt formulation of volumatic forces is given by

$$P = K^{\text{vol}}(S^{\text{init}} - S) - C^{\text{vol}}\dot{S}$$

where K^{vol} and C^{vol} denote elastic and viscous coefficients of volumatic deformation.

2.2.6 Particle-edge connection in coarse-fine model

We can apply coarse-fine approach to the geometric description of a 2D object. Figure 2.23 shows a simple 2D shape described by coarse-fine approach. This description consists of differently sized two triangles. Triangle T_0 consists of three edges E_0 , E_1 , and E_2 . Edge E_1 is connecting particles P_1 and P_3 . Triangle T_1 consists of three edges E_3 , E_4 , and E_5 . Edge E_3 is connecting particles P_3 and P_2 . Thus, we have the geometric description of the object listed in Table 2.2.

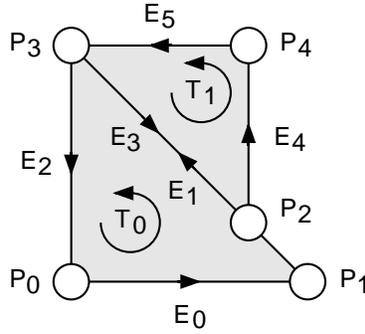


Figure 2.23: Example of coarse-fine description

Table 2.2: Coarse-fine description of 2D object by a set of triangles

edge	starting particle	end particle
E_0	P_0	P_1
E_1	P_1	P_3
E_2	P_3	P_0
E_3	P_3	P_2
E_4	P_2	P_4
E_5	P_4	P_3

triangle	edges		
T_0	E_0 +	E_1 +	E_2 +
T_1	E_3 +	E_4 +	E_5 +

In this description, particle P_2 should be on edge E_1 with an internal ratio predetermined in the initial shape of the object. In general, coarse-fine description of an object involves particles on edges. Each of the particles should be connected to an internally dividing point on a corresponding edge. Let particle P_i should be connected to an internally dividing point on edge E_a in coarse-fine description of an object. Particle P_i may lose its coincidence with the internally dividing point during the computation of the deformation. Then, particle P_i should be guided to the internally dividing point. This guidance can be performed by two artificial Voigt models of natural length 0 between the particle and the internally dividing point. One is parallel and the other is perpendicular to the edge. Let Q be the internally dividing point.

A particle-edge force to maintain particle P_i on internally dividing point Q is then given as follows:

$$\mathbf{f}_{i,a} = \{-Kd^n - Cd^n\}\mathbf{n}_a + \{-Kd^e - Cd^e\}\mathbf{e}_a$$

where $d^n = \overrightarrow{QP_i} \cdot \mathbf{n}_a$ and $d^e = \overrightarrow{QP_i} \cdot \mathbf{e}_a$ are signed distances along unit vectors \mathbf{n}_a and \mathbf{e}_a . This artificial force and its reaction force are applied to particle P_i and edge E_a to guide the particle to internally dividing point Q . Consequently, a set of particle-edge forces caused by force $\mathbf{f}_{i,a}$ defined in the above equation conserves the coarse-fine description of an object.

2.3 Particle-Based Modeling of 3D Object Deformation

Particle-based modeling can be extended to 3D deformation easily. Any 3D shape is described by a set of tetrahedra. Each tetrahedron consists of four particles, six edges, and four triangles.

Particle-edge forces in 2D deformation are extended to *particle-triangle forces* in 3D deformation. Force and its reaction are applied to a particle and its opposite triangle in a tetrahedron. Force applied to a triangle is equivalently split into three forces applied to the particles of the triangle. Thus, we can introduce topology maintenance forces in 3D deformation. Volumatic forces can also be introduced to 3D deformation.

Figure 2.24 demonstrates the computation of 3D deformation. A rheological food dough is formed by a rotating roller.

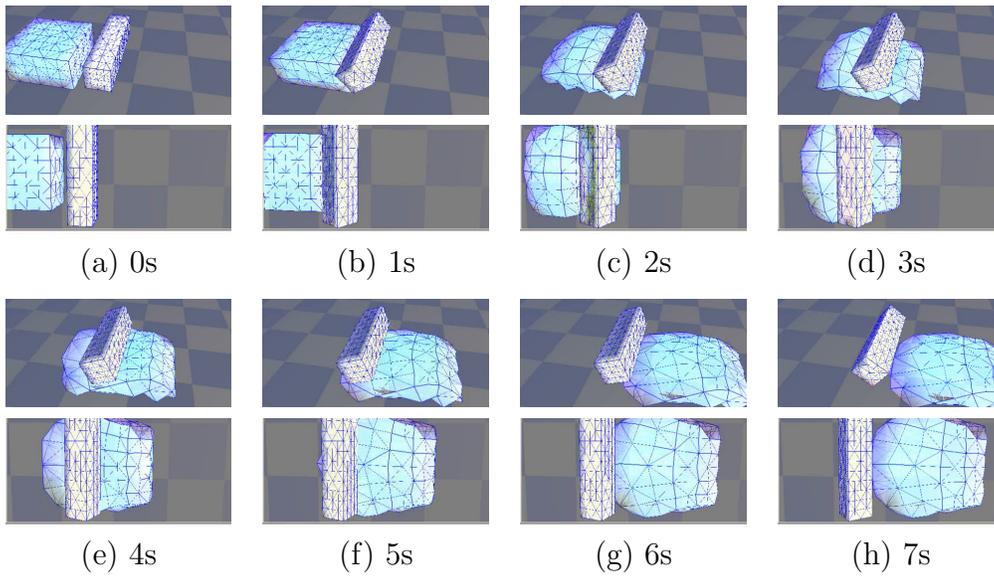


Figure 2.24: Forming process of dough

第2章 FPGAによる仮想レオロジー 物体のリアルタイム変形計算

2.1 緒言

近年，手術シミュレータを筆頭とし，医療分野やエンターテイメント分野において，リアルタイムでの物理シミュレーションが一般的なものとなりつつある．しかし，多くの場合，ハードウェアの演算能力が十分ではない．特に力覚提示を伴う場合には，1000Hz以上のサンプリングレートでの計算が必要となり，計算量が膨大な量となることから，複雑な物体の変形計算に難がある．また，10,000Hzでの力覚提示も登場しており，特に弾性が高く硬い物体において，より安定で高品質な力覚提示を実現するためには，1000Hzよりも高いサンプリングレートでの計算が必要であることが報告されている [17]．

物体変形シミュレーションを高速化する手法として，汎用 CPU の代わりに専用ハードウェアを用いる手法が考えられる．物体変形を高速に計算可能なハードウェアとして，DSP, ASIC, FPGA(Field Programmable Gate Array) が挙げられる．この内，ASIC と FPGA は，各演算回路が同時平行的に動作するため，本報告が対象とするような並列性の高いアルゴリズムとの親和性が高く，計算の高速化に適する．ただし，ASIC は，量産には適するものの，マスク作成にかかるコストが高いため少量生産には向かず，研究用途での製造は現実的ではない．それに対し，FPGA は回路の組み替えが可能な汎用の LSI であり，LSI を製造することなく低コストで専用ハードウェアを構築できる．また，回路を組みかえることで，物体変形モデルの変更等によるアルゴリズムの変更に柔軟に対応することが可能である．そこで，レオロジー物体のリアルタイム変形計算手法として，FPGA を用いた並列計算を提案する．

物理シミュレーションの分野では，天体の重力の計算に特化した計算機として，FPGA を用いた天文シミュレーション計算機 GRAPE-6 が構築されており，銀河形成シミュレーション等の各種天文シミュレーションに用いられている [6]．ただしこれは天体の計算に特化した計算機であり，物体の変形を扱うものではない．またこれはリアルタイムでの計算を目的としておらず，莫大な計算量を要するシミュレーションを実用的な処理時間で実行することに主眼を置いている．

物体の変形を高速に計算することが可能なモデルとしては、境界要素法 (BEM) が提案されており [8, 9]、また、有限要素モデルで物体変形の計算を高速化する手法として、アダプティブメッシュの導入が提案されている [4]。しかし、これらの手法では、物体が均一であるという仮定を前提としており、手術シミュレータ等における生体組織のモデリングに必要となりうる異方性のモデル化を考慮すると拡張性に難がある。

そこで、本章では、FPGA を用いることで大規模な物体の変形をリアルタイムに計算する手法を提案する。FPGA での変形計算に用いる力学モデルについては、前章で構築したモデルを用いる。物体変形のモデリングは、コンピュータグラフィックスおよびバーチャルリアリティの分野において盛んに研究されているテーマである。弾性物体の変形に関しては、初期の弾性理論の導入 [20, 21] を初めとして数々の研究がなされている。レオロジー物体の力学モデルとしては、剛体運動と物体変形を組み合わせたモデルが提案されている [22]。しかし、FPGA は複雑な計算には不向きであり、多数回の単純な計算に適する。そこで本章では、よりシンプルなモデルとしてバネ質点モデル [23] をベースに構築したレオロジー物体の動力学モデル [25] を用いる。

本章では、計算ハードウェアおよび物体変形モデルの両面から、リアルタイム変形シミュレーションの高度化を目指す。本研究は、力覚提示等の用途に応用できるリアルタイム計算可能なレオロジー物体変形シミュレーション技術の高度化を共通目的とする二つの研究テーマに大別される。柱となるテーマは、内部回路の組み換えが可能な LSI である FPGA を用いたレオロジー物体シミュレーションの高速化である。

2.2 並列計算向きの変形計算アルゴリズム

2.2.1 レオロジー物体の定義

図 2.1-(a) に示す初期形状を有する物体に外力を作用させると、図 2.1-(b) に示すように変形すると仮定する。弾性物体では、図 2.1-(c) に示すように、外力を解放したときの形状が初期形状に一致する。外力を作用させた形状と外力解放後の形状との差を、戻り変位、初期形状と外力解放後の形状との差を残留変位とよぶ。弾性物体では、戻り変位があり、残留変位はない。粘塑性物体では、図 2.1-(d) に示すように、外力を解放したときの形状が変形形状に一致する。すなわち、粘塑性物体では、残留変位があり、戻り変位はない。図 2.1-(e) に示すように、戻り変位と残留変位の両方を有する物体を、本研究では、レオロジー物体と定義する。

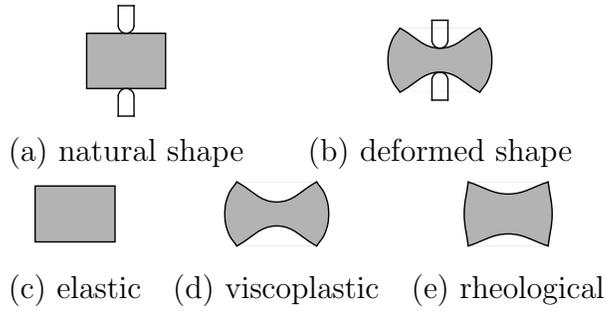


図 2.1: 弾性物体, 粘塑性物体, レオロジー物体

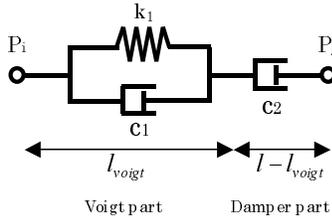


図 2.2: 三要素モデル

2.2.2 三要素モデル

レオロジー物体の性質を表現できる力学要素の中で最もシンプルなモデルとして、三要素モデルが挙げられる。このモデルは図 2.2 に示すように、戻り変位を持つフォークトモデルと残留変位を持つ単独のダンパを直列に接続したモデルである。

三要素モデルの定式化を行う。三要素モデルの長さおよび三要素モデルのフォークト部の長さをそれぞれ l , l_{voigt} とし、フォークト部の粘性係数および弾性係数を k_1 , c_1 , 単独ダンパ部の粘性係数を c_2 とする。単独ダンパ部の長さ l_{damper} は,

$$l_{\text{damper}} = l - l_{\text{voigt}} \quad (2.1)$$

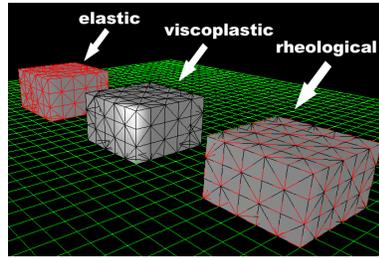
で表される。フォークト部が発生する力 f_{voigt} および単独ダンパ部が発生する力 f は,

$$f_{\text{voigt}} = -k_1(l_{\text{voigt}} - L) - c_1 \dot{l}_{\text{voigt}}, \quad (2.2)$$

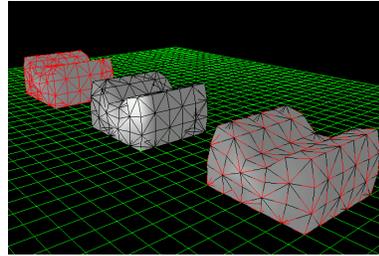
$$f = -c_2 \dot{l}_{\text{damper}} \quad (2.3)$$

となる。ここで、フォークト部に作用する力および単独ダンパ部にかかる力は等しいことから、(2.2) 式, (2.3) 式より、フォークト部の長さの微分 \dot{l}_{voigt} に関する式が導かれる。すなわち、定数 $A = -k_1/(c_1 + c_2)$ と $B = c_2/(c_1 + c_2)$ を導入すると,

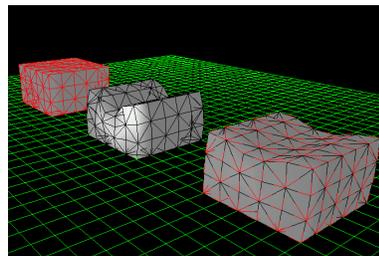
$$\dot{l}_{\text{voigt}} = A(l_{\text{voigt}} - L) + B\dot{l} \quad (2.4)$$



(a) initial



(b) deformed



(c) stationary

図 2.3: 三要素モデルによる粘弾性物体，粘塑性物体，レオロジー物体の変形シミュレーション

となる．

三要素モデルはフォークトモデル，マックスウェルモデル，単独のダンパを包含するものであり，パラメータに応じて，それらの力学要素と同等の性質を表現することができる．すなわち，完全弾性物体，レオロジー物体，粘塑性物体を連続的に表現することが可能である．

三要素モデルを用いたレオロジー物体の変形シミュレーションの例を図 2.3 に示す．図 2.3-(a) が初期状態，図 2.3-(b) が変形状態である．そして，図 2.3-(c) に示すように，外力解放後の変形状態が，弾性物体，レオロジー物体，粘塑性物体で異なる．三要素モデルを用いることで，パラメータ k_1, c_1, c_2 に応じ，弾性物体，レオロジー物

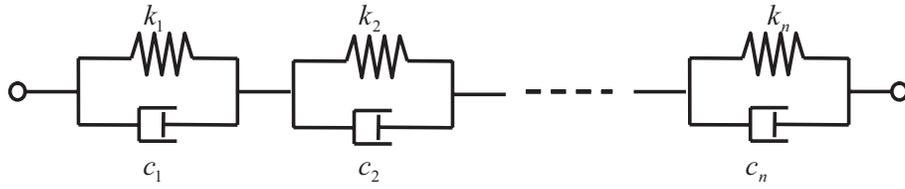


図 2.4: 一般化フォークトモデル

体, 粘塑性物体の性質を連続的に表現することができる.

2.2.3 一般化フォークトモデル

レオロジー物体を包含し, さらに多様な柔軟物を扱うことのできるモデルとして一般化フォークトモデルが挙げられる.

図 2.4 が一般化フォークトモデルの模式図である. 図中において, n 番目のフォークト部のパラメータは, 弾性係数 k_n , 粘性係数 c_n とする. また, システム変数として, n 番目のフォークト部の変位 $\Delta\epsilon_n^{\text{voigt}}$ を用いる. モデル全体の変位を $\Delta\epsilon$ とする. すなわち,

$$\Delta\epsilon = \sum_n \epsilon_n^{\text{voigt}}, \quad (2.5)$$

$$\dot{\Delta\epsilon} = \sum_n \dot{\epsilon}_n^{\text{voigt}} \quad (2.6)$$

である.

また, 各フォークト部が発生する力は, 全て同一であり,

$$f^{\text{gv}} = -k_n \Delta\epsilon_n^{\text{voigt}} - c_n \dot{\epsilon}_n^{\text{voigt}} \quad (2.7)$$

で表せる. (2.6), (2.7) 式より, システム変数の微分であるフォークト部変位速度 $\dot{\epsilon}_n^{\text{voigt}}$ を消去すると, 力 f^{gv} は,

$$f^{\text{gv}} = \frac{-\dot{\epsilon} - \sum_n \frac{k_n}{c_n} \Delta\epsilon_n^{\text{voigt}}}{\sum_n \frac{1}{c_n}} \quad (2.8)$$

となる. また, (2.8) 式で導出した f^{gv} から, n 番目のフォークト部の変位速度 $\dot{\epsilon}_n^{\text{voigt}}$ は,

$$\dot{\epsilon}_n^{\text{voigt}} = \frac{-f^{\text{gv}} - k_n \Delta\epsilon_n^{\text{voigt}}}{c_n} \quad (2.9)$$

で表される.

2.3 変形計算アルゴリズムのFPGA実相

FPGAを用いた仮想レオロジー物体変形計算ハードウェアを設計し、レオロジー物体変形シミュレーションの高速化を実現した。本章では、今回構築した仮想レオロジー物体変形計算システムについて述べる。

2.3.1 仮想レオロジー物体の変形モデル

FPGAは多数回の単純な繰り返し計算に適しており、これを効率的に演算することができる。しかし、複雑な計算には不向きであり、高速化に適さない。そこで、今回はシンプルな物体変形モデルとして、質点を離散的に配置し、質点間を一次元的な力学モデルで結ぶパーティクルベースモデルを用いた。また、レオロジーの性質を表現するための力学モデルとして、前述の三要素モデルを用いた。二次元物体および三次元物体は、図2.5に示すように、この三要素モデルを格子状に配置することで表現する。この格子は三角形または四面体の集合であり、各頂点に質点を置き、その質点間を結ぶ稜線に三要素モデルを配置し、質点間を接続することで、二次元もしくは三次元のレオロジー物体の変形を表現する。

質点 P_i を始点とする三要素モデルの集合を R_i 、質点 P_i を終点とする三要素モデルの集合を S_i で表す。稜線に向きを付け、第 k 稜線の始点から終点に向かう単位ベクトルを e_k と表す。また、第 k 稜線の三要素モデルの内力を f_k とする。このとき、集合 R_i に含まれる三要素モデル E_k が、質点 P_i に加える力は $f_k e_k$ に一致する。また、集合 S_i に含まれる三要素モデル E_k が、質点 P_i に加える力は $-f_k e_k$ に一致する。したがって、質点 P_i の運動方程式は、

$$m_i \dot{v}_i = f_{R_i} + f_{S_i} + F_i^{\text{ext}} \quad (2.10)$$

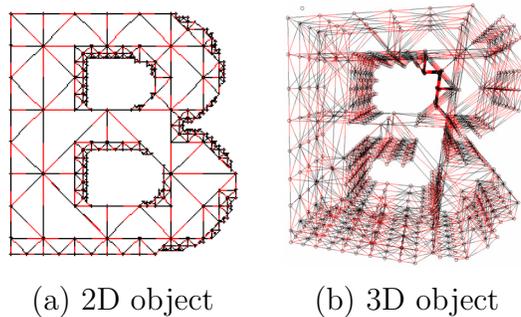


図 2.5: トラス構造による形状表現の例

となる。ただし, m_i は質点 i の質量, v_i は質点 i の速度,

$$f_{Ri} = \sum_{k \in R_i} f_k e_k, \quad f_{Si} = - \sum_{k \in S_i} f_k e_k$$

であり, F_i^{ext} は, 質点 P_i に作用する外力である。物体モデルの運動方程式は, 2.4)(2.10) 式で与えられる。仮想レオロジー物体の変形は, 運動方程式を数値的に解くことによって計算する。本研究における回路では, オイラー法を用いて計算する。

2.3.2 計算回路の構成

本節では, システム全体の概要および, 仮想レオロジー物体変形計算回路の各モジュールについて述べる。

システム構成

システムはPCとFPGA搭載PCIボードによって構成される。本研究では, FPGAとして Virtex-II XC2V6000 を用い, FPGA搭載PCIボードとして, 東京エレクトロニクス株式会社製の PC-BD-PCI2DVI を用いる。ただし, 他のFPGA搭載PCIボードでもほぼ同一の回路を動作させることが可能であり, 同様のシステムが構築できる。

変形シミュレーションの実行中, FPGAは単独で仮想レオロジー物体の変形計算を行い, PC側ではPCIバスを介してFPGAボードから画面表示のためのデータを取得する。力覚提示デバイスをシステムに追加する場合, デバイスの制御はPC側で行うことになる。ただし, 現行のPCIバスのデータ転送能力では, 1msごとに仮想物体の全データをPC側に転送するのは困難である。そのため, 1msごとに転送するのは力覚提示に必要な局所的な少数の点のみとする。グラフィック表示に必要な物体表面全体の質点の座標データは, より低速な, 画面のリフレッシュレートに合わせて転送すればよく, 33msをかけて転送する。

仮想レオロジー物体変形計算回路

FPGAに実装する物体変形計算回路は, ハードウェア記述言語 VerilogHDL を用いて設計した。今回設計した回路の概要を図 2.6 に示す。

回路は, i) 三要素モデル計算モジュール (Three-element model module; TEM), ii) 質点運動計算モジュール (mass particle module; MPM), iii) RAMモジュール, iv) PCIコアの4つのブロックからなる。PCIコアは, PCIバスを介してPCと相互にデータ転送を行うための回路であり, Xilinx社が提供する既存の回路を用いる。

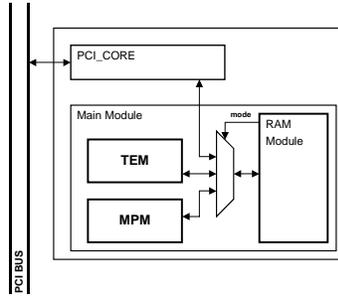


図 2.6: レオロジー物体変形計算回路概要

TEM は入力ポートから，第 k 稜線の始点 P_i の位置 $x_i(t)$ と速度 $v_i(t)$ ならびに終点 P_j の位置 $x_j(t)$ と速度 $v_j(t)$ ，稜線に対応する三要素モデルのフォークト部の長さ $l_k^{\text{voigt}}(t)$ を取得し，出力ポートに $f_k(t)e_k(t)$ と $l_k^{\text{voigt}}(t+h)$ を出力する．ただし，ここで h はオイラー法における刻み時間である．力 $f_k(t)e_k(t)$ は逐次，RAM module 内に蓄積されていき，TEM による全ての三要素モデルの計算が終わると，質点 P_i に作用する内力ベクトルの総和 $f_i^{\text{sum}}(t) = f_{Ri}(t) + f_{Si}(t)$ が得られる．

MPM は入力ポートから質点 P_i に作用する力ベクトルの総和 $f_i^{\text{sum}}(t) + F_i^{\text{ext}}$ および，質点 P_i の位置 $x_i(t)$ ，速度 $v_i(t)$ を取得し，出力ポートに $x_i(t+h)$ と $v_i(t+h)$ を出力する．

すなわち，TEM と MPM を交互に動作させることで変形シミュレーションが進行する．ただし，TEM と MPM を並列に動作させることはできない．今回の計算回路は，三要素モデル計算モード，質点計算モード，PCI バスデータ転送モードの三つの状態を遷移する．回路は全てパイプライン回路となっており，計算に要する時間は，質点の数と三要素モデルの数の合計に比例する．回路における数値の形式としては 16bit 固定小数点を用い，この 16bit の内，符号部，整数部，小数部にそれぞれ 1bit，7bit および 8 bit を割り当てる．

TEM

TEM の詳細を図 2.7 に示す．図中の回路 1 では三要素モデルの両端に繋がる質点 P_i, P_j の相対位置 $x_k^{\text{rel}}(t) = x_j(t) - x_i(t)$ を計算する．回路 2 では，質点 P_i, P_j の相対速度 $v_k^{\text{rel}}(t) = v_j(t) - v_i(t)$ を計算する．回路 3 では，三要素モデルの長さの逆数 $l_k^{\text{inv}}(t)$ を求める．すなわち，

$$l_k^{\text{inv}}(t) = \frac{1}{\sqrt{x_k^{\text{rel}}(t) \cdot x_k^{\text{rel}}(t)}} \quad (2.11)$$

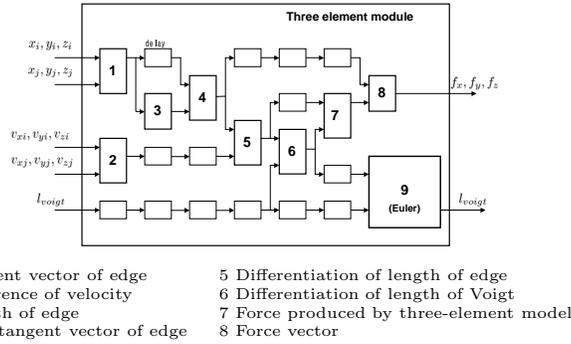


図 2.7: 三要素モデル計算モジュール

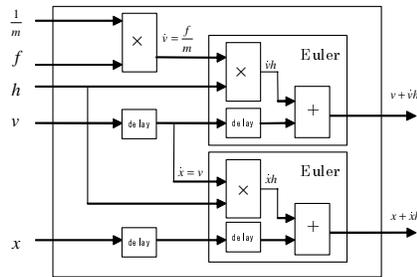


図 2.8: 質点運動計算モジュール

である．ここで含まれる平方根の演算には、CORDIC アルゴリズムを用いる．回路 4 では三要素モデルの単位方向ベクトル

$$\mathbf{e}_k(t) = l_k^{\text{inv}}(t) \mathbf{x}_k^{\text{rel}}(t) \quad (2.12)$$

を求める．回路 5 では稜線の長さの微分 $\dot{l}_k(t) = \mathbf{e}_k(t) \cdot \mathbf{v}_k^{\text{rel}}(t)$ を計算する．回路 6 では、(1.5) 式を用いて、フォークト部の長さの微分 $\dot{l}_k^{\text{voigt}}(t)$ を計算し、回路 7 では、(1.3) 式で示される三要素モデルの発生する力の大きさ $f_k(t)$ を計算する．回路 8 で、三要素モデルの発生する力ベクトル $\mathbf{F}_k(t) = f_k(t) \mathbf{e}_k(t)$ を求める．回路 9 では次のステップにおけるフォークト部の長さ $l_k^{\text{voigt}}(t+h)$ を求める．すなわち、

$$l_k^{\text{voigt}}(t+h) = l_k^{\text{voigt}}(t) + h \dot{l}_k^{\text{voigt}}(t) \quad (2.13)$$

を計算する．

MPM

MPMの詳細を図2.8に示す。MPMは運動方程式計算部と、二つのオイラー法計算モジュールで構成される。オイラー法モジュールは、質点に関するシステム変数の微分、すなわち質点速度の微分 \dot{v}_i および質点位置の微分 \dot{x}_i を入力とする。今回、全質点の質量は同じ値としており、その逆数 m_{inv} を用いて、速度の微分 $\dot{v}_i(t)$ を、

$$\dot{v}_i = m_{inv} f_i^{sum} \quad (2.14)$$

で計算する。また、位置の微分は $\dot{x}_i(t) = v_i(t)$ で求める。

RAM Module

RAM内の値に加算を行う場合には、通常は読み出しと書き込みを交互に行う必要があるが、今回のBlock-RAMは二つのポートを同時に使用できるものとなっており、二つのポートをそれぞれ、RAM内の値の読み込みと、それに加算した値の書き込みに使用することで、RAM内の値への加算処理をパイプライン化することができる。RAM moduleは、三要素モデルおよび質点に関するパラメータおよび、変数 $x_i, v_i, f_{Ri}, f_{Si}, F_k^{ext}, l_k^{voigt}$ を保持する。RAM moduleは主にFPGA内蔵BlockRAMで構成されており、三要素モデルの両端点に発生する力ベクトルを別々のBlockRAMに蓄積し、読み出し時に合計することで、質点 P_i に作用する力 f_{Ri} と f_{Si} を求める。これは同時に二つの質点のアドレスに対して、RAM内の値に加算処理を行うことができないためである。

RAM moduleは最小構成において、30個のBlockRAMを要する。まず、BlockRAMの容量は16bit × 1024wordであるため、 n_m 個の値の格納に必要なBlockRAMの個数は $n_{mr} = [(n_m - 1)/1024] + 1$ であり、質点に関するデータである x_i, v_i, f_{Ri}, f_{Si} および F_i^{ext} を格納するのに必要なBlockRAMの数はそれぞれ $3n_{mr}$ となる。また、 n_e 個の値の格納に必要なBlockRAMの個数は $n_{er} = [(n_e - 1)/1024] + 1$ であり、三要素モデルに関するデータである l_k^{voigt} を格納するのに必要なBlockRAMの数は n_{er} となる。また、三要素モデルの両端に繋がる質点の番号を格納するのに必要なBlockRAMの数は $2n_{er}$ である。

結局、仮想レオロジー物体の格納に必要なBlock RAMの総数 n_{br} は、

$$n_{br} = 15n_{mr} + 3n_{er}. \quad (2.15)$$

となる。立方体状の物体において三要素モデルの総数は質点の総数の約5倍となるため、 n_{er} は n_{mr} の5倍に設定する。したがって、 $n_{mr} = 1$ である最小構成1セット分のRAMは30となる。また、144個の内蔵BlockRAMを持つXC2V6000には、最大

4 セットの RAM module が格納できる。すなわち，4096 個の質点データと 20480 個の三要素モデルのデータが格納可能である。

回路は 33MHz で動作するため一つの BlockRAM は低速である。しかし，三要素モデル計算モードと質点計算モードのそれぞれにおいて，BlockRAM 計 15 個の各 2 ポートに同時にアクセスするため，バス幅 480bit の信号線が常時稼動し，ロスのない 1.98Gbyte/秒でのデータ転送が行われる。

2.3.3 誤差の抑制

今回の回路には，固定小数点を用いているため，正常にシミュレーションを行うためには計算誤差の抑制が一つの課題となる。仮想物体変形シミュレーション特有の問題として，オイラー法における刻み時間の演算がある。たとえば，サンプリングレート 1000Hz の力覚提示を想定すると，刻み時間 h は 0.001 という，小さな値となるため，本研究で用いる 16bit 固定小数点ではオイラー法の計算誤差がクリティカルな影響を生じ，正常にシミュレーションを行うことができない。そこで誤差を抑制するために，スケーリング，丸め処理を導入するとともに，静的に小数点位置を最適化する。

スケーリング

固定小数点における計算精度を向上するために，一般に知られる単位別スケーリング法を用いる。オイラー法の刻み時間の値には，一般にミリ秒単位の小さな値を用いるため，誤差の要因になりやすい。前述のように，オイラー法の刻み時間 h を 0.001 としたとき，これは小数部 8bit の固定小数点で表現可能な最小値を下回っているため，固定小数点に変換することができない。そこで本研究では時間に対して，単位の変換を行う。すなわち，単位の変換のための変換係数を r とおき，時間の単位 s を， rs に変換する。これに伴い，各種物理量は表 2.1 に示す単位となる。これにより， r の値を 0.1 と選ぶと，刻み時間 0.001 は，16bit 固定小数点で +0000000.00000010 と表される。すなわち，刻み時間を固定小数点に変換することが可能となる。

丸め処理

丸め処理を行わずに演算を行うと，出力結果から端数が切り捨てられるため，出力値が負方向に偏るという問題が発生する。仮想物体の変形シミュレーションでは，オイラー法計算においてその影響が如実に現れ，物体が際限なく負方向に変形し続けるという問題が発生する。そこで，これを改善するために演算結果の端数に対する丸め

表 2.1: 単位変換表

	S	RS
Time:	$t[s]$	$t/r[rs]$
Elasticity:	$k[\text{kg}/s^2]$	$kr^2[\text{kg}/rs^2]$
Viscosity:	$c[\text{kg}/s]$	$cr[\text{kg}/rs]$
Force:	$f[\text{kgm}/s^2]$	$fr^2[\text{kgm}/rs^2]$
Velocity:	$v[\text{m}/s]$	$vr[\text{m}/rs]$

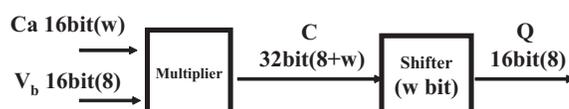


図 2.9: 固定小数点位置を最適化した定数乗算回路

処理を導入する。

丸め処理としては、四捨五入丸めや JIS 丸め等の手法が存在する。四捨五入丸めにおいて 0.5 などの中間値を丸める際に発生する統計的偏りを抑制したものが JIS 丸めである。JIS 丸めは四捨五入丸めよりも高精度であるが、本研究では回路規模と精度の兼ね合いから四捨五入丸めを採用する。オイラー法の乗算部に四捨五入丸めを導入することで、前述の負方向の変形が生じなくなる。

定数乗算処理における静的な小数点位置の最適化

乗算操作によって倍のビット幅の出力値が得られることを利用し、図 2.9 のような構成の演算器における計算プロセスの一部で小数点位置を変更することができる。すなわち、入力変数 V_b と出力値 Q は既定の小数点位置を持つようにしつつ、定数値 C_a および計算過程の値 C における小数点位置を変更することで計算精度が向上する。図の回路において、定数値 C_a の小数点位置を w とする。このとき、定数 C_a と変数 V_b の積 C の小数点位置は $8 + w$ である。次に、32bit 固定小数点 C から 16bit 分を取り出し、出力値 Q を得る。このとき、取り出される 16bit 分のデータが、 C の 32bit の範囲に収まるようにするためには、

$$w \leq 16 \quad (2.16)$$

を満たす必要がある。また、定数値 c_a が格納時にオーバーフローしないようにするためには、

$$|c_a| \leq 2^{15-w} - 2^{-w} \quad (2.17)$$

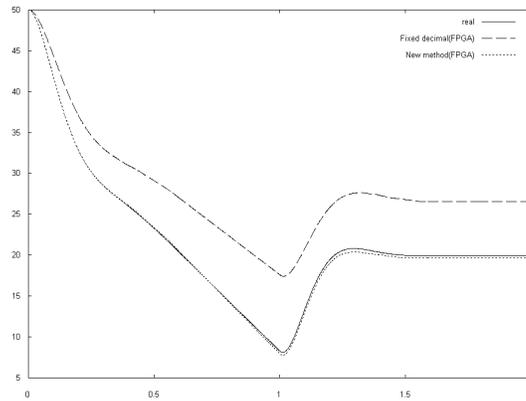


図 2.10: 小数点位置最適化による三要素モデル変形推移の変化

を満たす必要がある．よって, (2.16) 式および (2.17) 式から定数値における最適な小数点位置を,

$$w_{\text{mod}} = \begin{cases} 14 - \lfloor \log_2 |c_a| \rfloor & (14 - \lfloor \log_2 |c_a| \rfloor \leq 16) \\ 16 & (14 - \lfloor \log_2 |c_a| \rfloor > 16) \end{cases} \quad (2.18)$$

と決定することができる．すなわち, 乗算結果の 32bit 値から 16bit の出力値を取り出す前に w_{mod} ビットのシフトを行えばよい．定数値が不変である時, w_{mod} も一定であり, 最適な小数点位置が静的に確定する．この手法を用いることで, 回路規模を増加させることなく演算精度を向上することができる．

通常的小数部 8bit の固定小数点において, 刻み時間 0.01rs を固定小数点に変換すると, 0.0078125 となり, 21.9%の誤差を生じる．それに対し, 小数点位置を最適化すると変換後の値は 0.0099793 となり, 誤差は 0.2%にまで抑制される．

この手法を MPM に適用し, 単一の三要素モデルに外力を加えた時の変形シミュレーション結果を図 2.10 に示す．シミュレーション条件として, 刻み時間 $h = 0.001$, 質点の質量 $m = 1$, 三要素モデルパラメータ $k_1 = 250$, $c_1 = 20$, $c_2 = 100$ とし, 時刻 0 から 1 の間, 大きさ 3000 の力を作用させる．変換係数 r の値は, 0.1 とする．このシミュレーション例は MPM の計算精度を測るものであり, TEM が受け持つ部分は PC を用いて浮動小数点で計算し, 質点に関する計算を FPGA 内の MPM で行っている．

小数点位置を固定した通常固定小数点では, 平均 12.6%の誤差が生じているのに対し, 小数点位置を最適化した場合は, 0.407%まで誤差が減少しており, 全計算を浮動小数点で行った場合の変形推移に対し遜色のない結果になっている．

表 2.2: レオロジー物体変形計算回路の回路規模

	flip-flops	slices	multipliers
All modules	8,720	6,875	25
TEM	6,880	4,144	16
MPM	1045	640	9

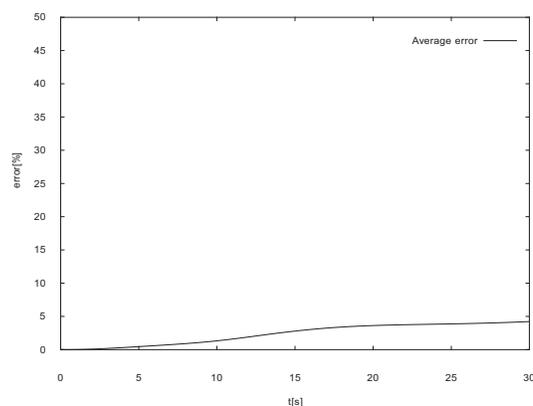


図 2.11: 三次元物体変形過程における平均位置誤差

表 2.3: 各軸質点数 $9 \times 9 \times 9$ の物体における FPGA と PC の処理時間の比較

	time[ms]
PC(PentiumIV 1.7GHz)	1.14
FPGA(XC2V6000)	0.138

2.3.4 FPGA 実装

今回設計した仮想レオロジー物体変形計算回路を FPGA に実装し、計算システム実機で仮想レオロジー物体の変形シミュレーションを行った。作成した回路の規模を表 2.2 に示す。この回路は、XC2V6000 の回路の 20% を占めている。

PC による計算結果と比較するため、三次元物体の変形シミュレーションを PC と FPGA で行った。対象となる物体は、各軸 $9 \times 9 \times 9$ の質点を持つ立方体状の物体であり、この物体に対し、物体上面の二つの角に外力を加えた。パラメータやシミュレーション条件等は FPGA と PC でどちらも同じものとし、外力以外の条件については前節のシミュレーションと同じとした。外力については、 $F_0^{\text{ext}} = [6.0, 30.0, 6.0]^T$ および $F_1^{\text{ext}} = [-30.0, -15.0, -30.0]^T$ を時刻 0 から時刻 30 までの間、物体上面の二点に加えた。

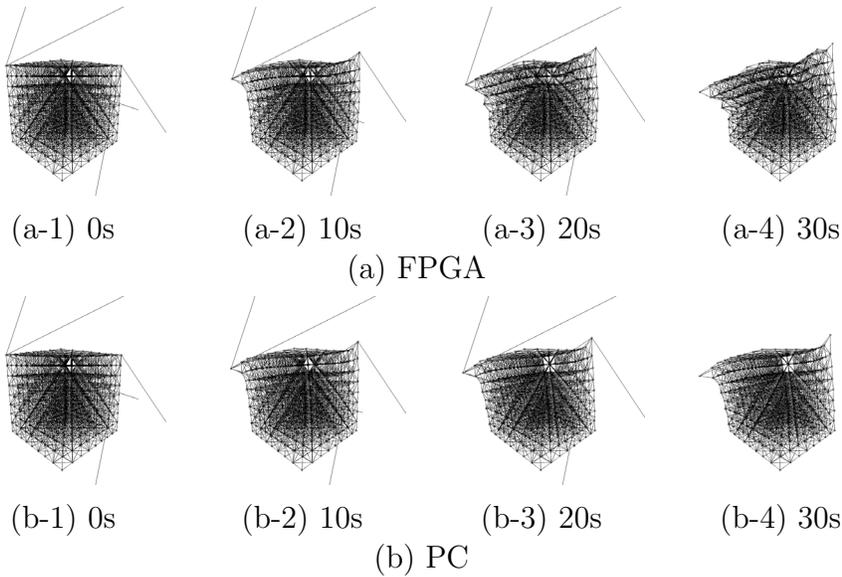


図 2.12: 仮想レオロジー物体の変形形状の比較

FPGA および PC それぞれによる変形シミュレーションによって得られた物体の変形形状を図 2.12 に示す．また，変形過程における各時刻での質点の位置誤差の平均を図 2.11 に示す．誤差には蓄積性があり，30s の時には位置誤差の平均は物体の初期幅に比して 4.22%まで増大している．

PC と FPGA のそれぞれにおいて，画面表示等の処理を除く，純粋な物体変形計算に要する時間を測定すると，表 2.3 が得られた．FPGA 実機において，PC の 8.26 倍の処理能力が実現されている．

2.3.5 実現可能な処理能力

複数回路による並列計算

TEM, MPM, RAM module で構成される Main module を複数個用いて，並列計算を行うことが可能である．物体を構成する質点群を回路の数と同数のグループに分割し，それぞれの Main module の BlockRAM に各グループの質点データを格納することで並列化が可能となる．MPM による質点に関する計算は完全並列であり，回路の数に比例した処理能力を得ることができる．一方，TEM における三要素モデルの計算は，完全並列ではない．これは質点グループ間にまたがる三要素モデルが存在するためである．グループ間にまたがる三要素モデルを計算する際には，TEM に対し，同時に二つの Main module 内の RAM module からデータを入力する必要があり，そ

の時，二つの Main module の片方にしかデータを入力することができないので，片方の Main module において 1 クロック分のロスを生じる．

処理能力の試算

本研究が対象とするアルゴリズムは条件分岐を持たず，パイプラインストールを生じないため，FPGA による変形シミュレーションに要する時間を明確に予測することができる．一回のループにおいて TEM が要するクロック数 c_{te} は，

$$c_{te} = \frac{n_{es} + 2n_{et}}{P_m} + d_{te} \quad (2.19)$$

となる．ここで， n_{es} は，一つのグループ内の質点間を繋ぐ三要素モデルの総数であり， n_{et} は二つのグループにまたがる三要素モデルの総数である．また， P_m は計算回路の数であり， d_{te} は TEM の計算回路のパイプライン段数である．すなわち，TEM に最初にデータを入力してから， d_{te} クロック後に最初の演算結果が出力される．

また，同様に MPM が要するクロック数 c_{mm} は以下の式で与えられる．

$$c_{mm} = \frac{n_m}{P_m} + d_{mm} \quad (2.20)$$

ここで， n_m は質点の総数であり，また， d_{mm} は MPM のパイプライン段数である．

FPGA の動作周波数を r_c とすると，総処理時間 t_s は，

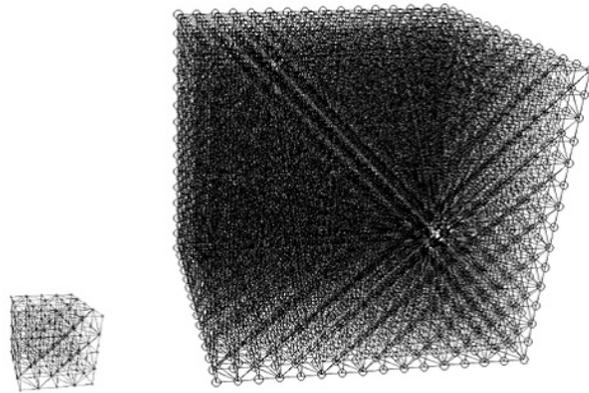
$$t_s = \frac{c_{te} + c_{mm}}{r_c} \quad (2.21)$$

となる．

本研究の回路では， $d_{te} = 90$ ， $d_{mm} = 8$ ， $r_c = 33MHz$ であり，各軸質点数 $9 \times 9 \times 9$ の物体の 1 ループあたりの処理時間は試算より，0.137ms となる．実機における同じ物体の処理時間は 0.138ms であり，精度良く処理時間が予測されている．

FPGA に格納可能な仮想物体の規模は，2.3.2 節で述べた通り，質点総数 4096，稜線総数 20480 であり，この制限内での最大クラスの直方体は各軸質点数 $15 \times 15 \times 16$ ，質点総数 3600，稜線総数 19531 のものである．本研究における回路の処理時間は，この物体において 0.704ms と試算される．すなわち，サンプリングレート 1000Hz での計算が可能である．今回構築した回路は，FPGA による仮想レオロジー物体変形シミュレーション回路の最小構成であり，1) Main module の並列化，2) 動作周波数の引き上げ，3) PCI バス転送処理の並列化，による高速化の余地がある．

2.3.2 節で述べたように，XC2V6000 には 4 セットの RAM module が格納可能である．Main module は 1 セット分の BlockRAM を要するため，並列可能な Main module



(a) PC ($5 \times 5 \times 5$) (b) FPGA ($15 \times 15 \times 16$)

図 2.13: サンプリング時間 10kHz で計算可能な最大規模の物体

の数は, XC2V6000 において最大 4 となる. また, 今回の変形計算回路が占める回路規模は XC2V6000 の 20% の領域であり, 4 個の Main module を FPGA 上に実装し, Main module を並列化することが可能である. また, 回路の動作周波数は, XC2V6000 に内蔵される DCM(Digital Clock Manager) を用いて, 66MHz に逡倍することができる. PCI バス転送処理の並列化は, 各 RAM module に $3n_{mr}$ 個の RAM を新たに追加し, 質点計算モードの時に出力される質点位置を記憶させることで可能となる. データの転送は稜線計算モード時に並行的に行うことができる. これを追加した場合でも, XC2V6000 には 4 セットの RAM module が格納可能である.

これらを実現した場合, 各軸質点数 $15 \times 15 \times 16$ の物体を 0.0920ms で処理することが可能になると試算される. すなわち, サンプリングレート 10,000Hz での変形シミュレーションが可能である. 比較として, 10,000Hz で計算可能な物体を図 2.13 に載せる. 図 2.13-(a) の物体の PC における処理時間は実測値で 0.0948ms であり, 10,000Hz で計算可能なほぼ限界のサイズの物体である. また, PC で図 2.13-(b) の各軸質点数 $15 \times 15 \times 16$ の物体を計算したところ, 処理時間は実測で 6.92ms となった. これより, 現行のシステムで実現可能な計算能力は PentiumIV1.7GHz 搭載 PC の 75.2 倍と試算される.

一般に, CPU の進歩に比して主記憶メモリのアクセス速度の進歩は低速であり, 大規模物体のシミュレーションにおけるボトルネックとなりがちである. しかし, FPGA においては, 多数の RAM を並列動作させることで高速な計算を実現しており, 低速な RAM を用いても計算能力が RAM のアクセス速度に依存しないという特徴がある. これが今後の進歩におけるアドバンテージになる可能性がある.

2.4 結言

FPGA を用いた仮想レオロジー物体変形システムを構築し, PentiumIV 搭載 PC の 8.26 倍の速度での仮想レオロジー物体変形シミュレーションを実現した. 同時に, 同 PC の 75 倍の計算能力が現行のシステムで実現可能であることを試算より示した.

本研究における FPGA による仮想物体変形計算システムにおいて, 同一 FPGA 内への複数の Main module の配置による並列計算が可能であることを述べた. これらの回路は必ずしも同一の FPGA 内に配置する必要はなく, 原理上, 複数の FPGA で並列計算を行うことが可能である. この時, 扱える物体の規模は FPGA の総数に比例する. また, 物体規模によらず, 各グループに隣接するグループの数は 6 以下であり, FPGA 間の接続に必要な信号線の数は物体規模に関わらず一定以下で抑えられるため, 任意の数の FPGA を並列化できる. また, 処理時間も物体規模に関わらず一定以下に保たれる. すなわち FPGA の並列化により, 処理時間が明確に予測可能なリアルタイム物体変形シミュレーションが任意の物体規模で実現可能であると考えられる. ただし, FPGA による変形シミュレーションの課題として物体の大変形への対応が挙げられる. 本章で述べた回路は大変形に対応しておらず, 位相が崩れる可能性がある. この対処法としては, 位相保持のための力学要素を追加するか, もしくは本報告で述べる大変形に対応したモデルを FPGA 上に実装する必要がある.

その他, FPGA が効果を発揮する計算対象として, FEM による物体変形シミュレーションが挙げられる. FPGA は除算および平方根計算よりも乗算処理を得意としており, 大量の乗算処理を必要とする行列計算に適している. 物体全体の行列計算を必要とする通常の FEM は, 並列計算になじまない. しかし, 節点に生じる力の計算は局所的な小規模の行列で計算することが可能であり, また, 節点に質量を持たせる集中定数型の慣性行列を用いれば, 節点の位置と速度の計算にも物体全体の行列を要しない. すなわち, 本報告と同様に並列計算を行うことで, 任意の物体規模でのリアルタイムシミュレーションを行うことが可能であると考えられる.

FPGAによる仮想レオロジー物体のリアルタイム変形計算

立命館大学ロボティクス学科 友國誠至, 平井慎一

Real-time Computation of Rheological Deformation Using FPGA

Seiji TOMOKUNI and Shinichi HIRAI Department of Robotics, Ritsumeikan University

Abstract: Deformable soft objects such as food and tissue show both elastic and viscoplastic properties, and are referred to as rheological objects. We designed a computational hardware using an FPGA(Field Programmable Gate Array) and realized a system that computes the rheological deformation 18 times as fast as a PC with a Pentium IV 1.7 GHz. Our estimations show that an FPGA can compute the deformation 243 times faster than the PC.

1 はじめに

近年, 医療分野やエンターテインメント分野において, リアルタイムでの物理シミュレーションが一般的なものとなりつつある. しかし, 多くの場合, ハードウェアの計算能力が十分ではなく, 特に力覚提示を伴う場合には 1000Hz 程度のリフレッシュレートが必要となるため, 複雑な物体の変形計算に難がある状況である.

一方で, ハードウェアの分野では, FPGA と呼ばれる回路の組み換えが可能な LSI が登場し, 急激にその回路規模を増している. FPGA では各演算回路が同時並行的に動作するため, シミュレーション分野によく見られる並列計算アルゴリズムとの親和性が高く, シミュレーションの高速化に一定の効果を発揮する. 今回, 我々は FPGA 上に演算回路を実装し, PentiumIV1.7GHz 搭載 PC の 18 倍の速度での仮想レオロジー物体の変形計算を実現した. また, 現行の FPGA を用いて同 PC の 243 倍の速度での計算が可能であるという試算を得た.

2 レオロジー物体のモデル

レオロジー物体とは, 食品や生体組織, 粘土などに代表される柔軟物であり, 完全弾性物体と塑性物体の中間的な性質を有する物体である. レオロジーの性質を表現する一次元モデルとしては Fig.1 に示す三要素モデル [1] があり, フォークトモデルと単一のダンパを直列に接続したものである. 三要素モデルの長さおよび三要素モデルのフォークト部の長さをそれぞれ l, l_{voigt} とし, フォークト部の弾性係数および粘性係数を k_1, c_1 , ダンパ部の粘性係数を c_2 とし, 定数 $A = -k_1/(c_1 + c_2)$, 定数 $B = c_2/(c_1 + c_2)$ とおくと, フォークト部の比率の変化および三要素モデル

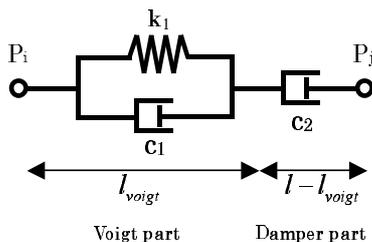


Fig. 1: Three element model

の発生する力は,

$$\dot{l}_{voigt} = A(l_{voigt} - L) + B\dot{l} \quad (1)$$

$$f = -c_2(\dot{l} - \dot{l}_{voigt}) \quad (2)$$

で表される.

三要素モデルの両端には質点 P_i, P_j が繋がっており, 物体の変形は三要素モデルから各質点に加わる合力による運動方程式に基づいて計算される. 物体は三角形要素あるいは四面体要素を組み合わせて表現し, 各要素の頂点に質点を配し, 頂点間を結ぶ稜線を三要素モデルとする.

3 レオロジー変形処理回路の概要

本研究で設計した変形計算回路の概要を Fig.2 に示す.

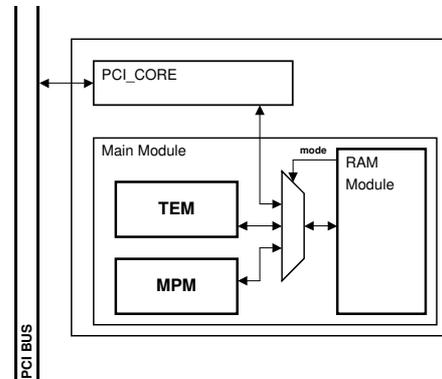


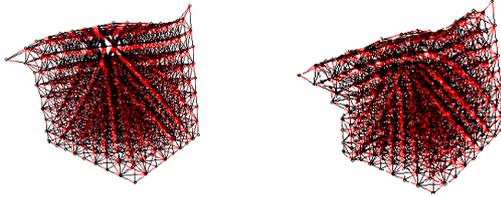
Fig. 2: Schematic computation module

図中において TEM は各稜線に配置された三要素モデルによって生じる力および三要素モデルフォークト部の変形を計算する回路であり (1),(2) 式の演算を行う. MPM は一般的な運動方程式に基づく質点の運動を計算する回路である. それぞれの回路はオイラー法の演算を含むパイプライン回路となっており, この二つの回路が交互に動作することで, 1ループ分のシミュレーションが行われる.

本研究では FPGA として Xilinx 社の VirtexII XC2V6000 を用い, PCI バスを介して XC2V6000 搭載ボードを PC と接続する. 回路は PCI バスのクロックと同期し, 周波数 33MHz で動作させる. また変数の値は 8bit 目を小数点位置とする 16bit 固定小数点で表現する.

Table 1: Size of Circuits

	flip-flops	slices	multipliers
All modules	8,720	6,875	25
TEM	6,880	4,144	16
MPM	1045	640	9



(a) PC (b) FPGA

Fig. 3: Deformed shape of rheological objects

Table 2: Computational time

	time[ms]
PC(PentiumIV 1.7GHz)	3.65
FPGA(XC2V6000)	0.20

固定小数点による誤差を減少させるため、四捨五入丸めを導入し、また静的に小数点位置を設定できる定数値の固定小数点位置を最適な値に設定し、局所的に小数点位置を変更することで回路規模の増大を抑えつつ、定数値の精度の向上を行った。設計した回路の回路規模を Table1 に示す。今回設計した回路は、XC2V6000 の使用可能 slice 数の 20 % を占めている。

4 動作結果

PC と FPGA で $9 \times 9 \times 9$ の質点を持つ仮想レオロジー物体の上部二箇所に外力を加えた場合の変形を計算し、結果を比較した。得られた変形形状を Fig.3 に示す。また、PC と FPGA のそれぞれで計算に要した時間を、Table2 に載せる。表より、FPGA が PC の 18.25 倍の計算速度を実現していることが分かる。FPGA の計算精度は PC より低く、物体の変形形状は異なるが、定性的には類似した形状が得られた。

5 実現可能な処理速度の試算

今回の回路には TEM と MPM を一つずつ実装した。しかし、質点を複数のグループに分けることで複数回路による並列計算も可能であり、その処理能力は試算から得られる。回路 TEM が一回のループに要するクロック数は、

$$c_{te} = \frac{n_{es} + 2n_{et}}{P_{te}} + d_{te} \quad (3)$$

である。 n_{es} , n_{et} , P_{te} , d_{te} はそれぞれ、同一のグループ間を接続するエッジの総数、二つのグループにまたがるエッジの総数、TEM の回路数、TEM のパイプライン段数である。回路 MPM が一回のループに要するクロック

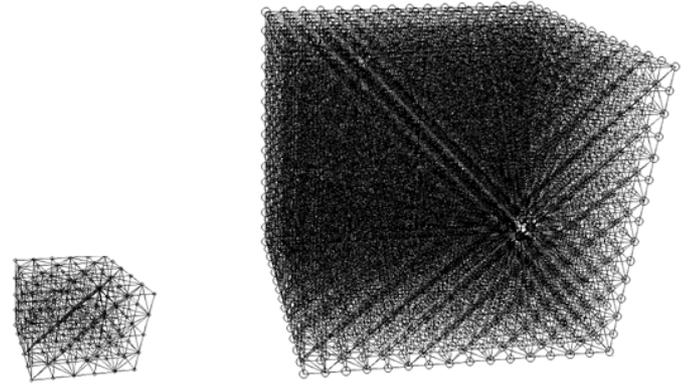
(A) PC ($5 \times 6 \times 6$) (B) FPGA ($15 \times 15 \times 16$)

Fig. 4: Maximum feasible virtual object size

数は、

$$c_{mm} = \frac{n_m}{P_{mm}} + d_{mm} \quad (4)$$

である。ここで、 n_m , P_{mm} , d_{mm} はそれぞれ質点の総数、MPM の回路数、MPM のパイプライン段数である。

これらより、1ループ当りの処理時間は、

$$t_s = \frac{c_{te} + c_{mm}}{r_c} \quad (5)$$

となる。ここで、 r_c は FPGA に与えるクロック周波数である。今回の回路では、XC2V6000 の RAM 容量の制約から格納可能な物体規模が最大 $15 \times 15 \times 16$ の質点を持つ物体までに限られるが、そのサイズの物体の処理に要する時間は、0.712ms である。したがって、1000Hz の更新頻度を必要とする力学提示用途では、PC と FPGA で扱える物体の規模はそれぞれ Fig.4 に示すものとなる。

今回は PCI バスと同期して 33Mhz で回路を駆動したが、VirtexII 内蔵回路による過倍で 66MHz 駆動を行うことが可能であり、また XC2V6000 には、TEM と MPM をそれぞれ 4 つずつ実装する余裕がある。これらを実装すると $15 \times 15 \times 16$ の物体の計算に必要な時間は 0.0955ms となり、PC で同サイズの物体を計算した場合の処理時間の実測値 23.2ms に対し、FPGA は 243 倍の速度で計算を行えることが試算より得られた。

6 おわりに

レオロジー変形計算回路を実際に動作させ、レオロジー変形計算の高速化を実現するとともに、試算から現行の FPGA で実現可能な計算速度を示した。今後の課題としては、力覚提示デバイスとの連携、体積効果および位相保持 [1] の実装が考えられる。

FPGA は並列アルゴリズム全般に利用することができ、他のシミュレーション分野においても今後の発展が期待される。

参考文献

- [1] 友國誠至, 杉山勇太, 平井慎一, ”実時間計算可能な仮想レオロジー物体の構築”, 日本バーチャルリアリティ学会論文誌, Vol.8, No.3, pp.247-254, 2003.

仮想レオロジー物体のモデリングおよび FPGA による実時間変形計算

Modeling of virtual rheological objects and realtime computation of their deformation on FPGA

学 友國誠至 (立命館大) 学 木村政文 (立命館大)
学 杉山勇太 (立命館大) 正 平井慎一 (立命館大)

Seiji TOMOKUNI, Ritsumeikan University, Nojihigashi1-1-1, Kusatushi, Shiga
Masafumi KIMURA, Ritsumeikan University, Yuuta SUGIYAMA, Ritsumeikan University,
Shinichi HIRAI, Ritsumeikan University

This paper describes the realtime computation for the deformation of virtual rheological objects on FPGA. Displaying tactile sensation requires to compute the deformation in high frequency. We will implement realtime computation of the deformation of large scale objects on an FPGA suitable for parallel computing. First, we will describe the simplified dynamic equations for computing the deformation of rheology objects. Second, we will outline the calculation of object deformation on an FPGA and will evaluate the performance of the calculation. Finally, we will discuss data transfer between a PC and an FPGA and will propose a better design for the realtime computation of the rheological deformation.

Key Words: Virtual Rheology, Deformation, Parallel Computing, FPGA

1 はじめに

近年, 力覚提示の分野における VR の進歩が目覚ましい. しかし仮想物体の対話操作において視覚提示が 30~60Hz 程度の時間分解能で行なわれるのに対し, 力覚提示を安定に行なうためには柔軟物において数 10~数 100Hz 程度の時間分解能が必要であり, その実現のためには仮想物体の変形計算を高速に行なう必要がある.

一方, ハードウェアの分野では FPGA と呼ばれる, 回路を書き換えることが可能な LSI が登場し, 年々回路規模を増している. この FPGA を用いて演算回路を設計することによって並列性の高いアルゴリズムを高速に計算することが可能である. 本報告では FPGA を用いた仮想レオロジー物体の実時間計算に関する報告を行なう.

2 レオロジー物体のモデル

レオロジー物体とは, 食品や生体組織, 粘土などに代表される柔軟物であり, 粘弾性物体と塑性物体の中間的な性質を示す物体である. レオロジー的性質を表現する一次元モデルとして Fig.1 に示す三要素モデルを用いる [1].

三要素モデルの長さおよび三要素モデルのフォークト部の長さをそれぞれ l, l_{voigt} とし, フォークト部の粘性係数および弾性係数を k_1, c_1 , ダンパ部の粘性係数を c_2 とする.

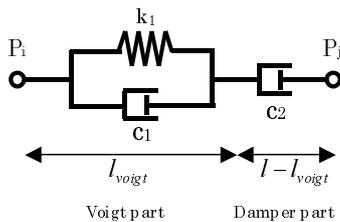


Fig. 1: Three element model

定数 $A = \left(\frac{-k_1}{c_1+c_2}\right)$, 定数 $B = \left(\frac{c_2}{c_1+c_2}\right)$ において計算式の簡略化を行なうと, フォークト部の比率の変化および三要素モデルの発生する力は (1) 式および (2) 式で表される.

$$\dot{l}_{voigt} = A(l_{voigt} - L) + B\dot{l} \quad (1)$$

$$f = -c_2(\dot{l} - \dot{l}_{voigt}) \quad (2)$$

三要素モデルの両端には質点 P_i, P_j が繋がっており, 物体の変形は三要素モデルが発生する力による各質点の運動方程式に基づいて計算する. 物体は三角形要素あるいは四面体要素を組み合わせて表現し, 各要素の頂点に質点を配し, 頂点間を結ぶ稜線を三要素モデルとする.

3 三要素モデル計算回路の概要

本研究では完全並列ではなく, パイプライン処理を行なう回路を数個並列に動作させることで変形計算を行なう. 三要素モデルパイプライン処理回路の概要を Fig.2 に示す.

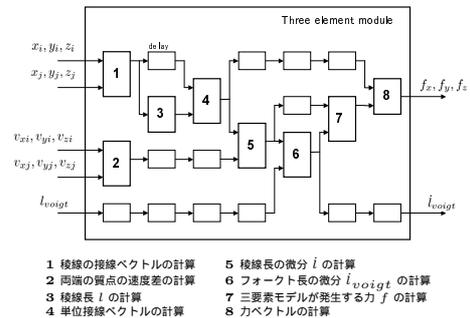


Fig. 2: Schematic computation module

本研究では FPGA として Xilinx 社の Virtex シリーズを対象とし, 開発言語として VerilogHDL を用いる. 三要素モデル計算回路は PCI バスによる PC との接続を想定する. 回路は PCI バスのクロックと同期し, 33MHz, 66MHz, 133MHz のいずれかの周波数で動作する. ここで回路の動作周波数と, 回

路規模およびレイテンシ (必要クロック数) はトレードオフの関係にある。設計した回路の配置配線および論理合成で得られた、動作周波数と回路規模およびレイテンシ、VirtexE シリーズの XCV2000E における回路使用率を Table1 に示す。

Table 1: Performance of module

動作周波数 [MHz]	77	125
レイテンシ	71	89
回路規模 [slices]	3259	3334
XCV2000E における回路使用率 [%]	16	17

4 三要素モデル計算回路の評価

ModelSim を用いて論理シミュレーションを行った。定数 k_1, c_1, c_2 はそれぞれ 0.5, 0.5, 2.3 とし, Table2 に示すデータを入力した。計算結果と理論値の比較を Table3 に示す。回路内では 16bit の固定小数点を用いているため出力結果と理論値との間に誤差が生じている。しかし、本研究ではリアリティよりも高速性を重視しているのでこれは許容できるものとする。

Table 2: Input data

x_i	y_i	z_i	v_{xi}	v_{yi}	v_{zi}	x_j	y_j	z_j	v_{xj}	v_{yj}	v_{zj}	l_{voigt}
1	2	3	4	5	6	6	5	4	3	2	1	7

Table 3: Error between result and theoretical value

	理論値	結果	絶対誤差	相対誤差 [%]
f_x	-1.3015	-1.3203	0.0188	1.4276
f_y	-0.7809	-0.7930	0.0121	1.5247
f_z	-0.2603	-0.2656	0.0053	2.0075
l_{voigt}	-3.8811	-3.8750	0.0061	0.1578

5 計算速度・実現可能な物体規模の試算

三要素モデルの計算以外の処理を全て PC 側で行った場合について、理論値による PCI バス転送時間も含む処理時間を試算したところ、PC と比較してほとんど高速化していないことが分かった。これは PCI バスの転送速度がボトルネックとなっているためである。しかし、運動方程式、ルンゲクッタ回路および質点・稜線データを格納するメモリを FPGA 側に実装することで、計算ループ毎に質点・稜線のデータを PCI バスで転送する必要がなくなりこのボトルネックが解消されようと考えられる。VirtexII シリーズの XC2V6000 で試算した場合、FPGA 内部の RAM には、格子数 $10 \times 10 \times 10$ (質点数 1331, 稜線数 6930) の仮想レオロジー物体のデータが格納可能であり、その物体に対して一つの計算回路を 66MHz で動作させた場合の時間分解能を試算すると 1176Hz となる。AthlonXP 2400+ を搭載した PC で処理時間を測定したところ、1000Hz の時間分解能で計算できる物体の規模は $6 \times 5 \times 5$ (質点数 150, 稜線数 661) であった。1000Hz の時間分解能での変形計算を対象とした場合、計算回路の FPGA 化により 10.5 倍の稜線数を持つ物体が扱えるようになると試算される。

6 位相問題

質点同士を一次元要素で繋ぐモデルでは、大変形の際に位相が崩れ、三角形や四面体などが反転することがある。その場合、元の形状に戻ろうとする復元力が逆向きとなるため、三角形や四面体が反転したままとなり、変形シミュレーションに不具合が生じる。この問題を解決するための位相保持の手法はいくつか存在する。我々のグループでは、質点と辺を繋ぐフォークトモデルによる位相保持アルゴリズムを PC 上に実装した [2]。また一次元要素によらないモデルとして、三要素モデルを用いた局所形状保持 [3] によるレオロジー物体および、離散要素モデルによる仮想物体の PC 上でのシミュレーションを行っている。シミュレーション例をそれぞれ、Fig.3, Fig.4 に示す。

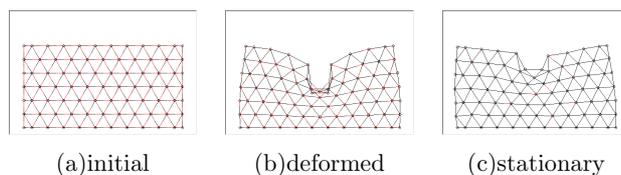


Fig. 3: Local shape maintaining model

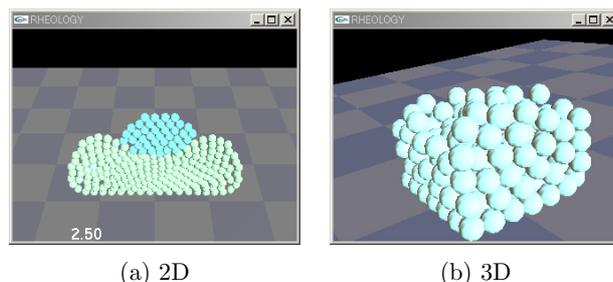


Fig. 4: Discrete element model

7 おわりに

本報告において三要素モデル計算回路のシミュレーションおよび計算速度・回路規模の試算を行い、FPGA による仮想物体の変形計算の高速化の可能性を示した。実機での動作については現在、小規模な FPGA における限定的なフォークトモデルの計算と PCI バスを用いたデータの送受信に成功しており、今後は大規模な FPGA 上への回路の実装および、実機によるリアルタイム変形計算を目指す。また、FPGA 上への位相保持アルゴリズムの実装可能性について検討する。

参考文献

- [1] 徳本真一, 平井慎一, "形状制御のためのレオロジー物体の変形モデリング", 日本ロボット学会誌, Vol.18, No.8, pp.87-94, 2001.
- [2] 木村政文, 杉山勇太, 友國誠至, 平井慎一, "仮想レオロジー物体の変形シミュレーション", 日本バーチャルリアリティ学会第7回論文集, pp183-186, 2002.
- [3] 宮崎慎也, 吉田俊介, 安田孝美, 横井茂樹, "局所形状保持に基づく仮想弾性物体モデルの提案", 電子情報通信学会論文誌 A, Vol.J82-A, No.7, pp.1148-1155, 1999.

第3章 視触覚提示のための仮想レオロジー物体のキャリブレーション

3.1 緒言

本章では、視触覚提示のための仮想レオロジー物体のキャリブレーションについて述べる。パーティクルベースモデルは、大変形に対応でき、連続体モデルと比較すると計算時間が短いという長所がある反面、モデル内の多数の力学パラメータが複雑に関連しており、モデル同定が困難であるという欠点を有する。連続体モデルでは、ヤング率や体積弾性率あるいは粘性率という、力学的に明快なパラメータを用いているのとは、対照的である。そこで本章では、レオロジー物体の三次元変形形状と物体に作用する力を計測するシステムを構築し、randomized algorithm あるいは genetic algorithm を用いて、パーティクルベースモデルの実際のレオロジー物体の計測値から、力学パラメータを同定する手法を確立する。この randomized algorithm は、もともと経路計画で開発された手法である。物体の計測値から力学パラメータを同定する過程を、パラメータ空間内の経路計画とみなすことにより、仮想レオロジー物体の力学パラメータを同定する。

レオロジー物体を表現する3つのモデルの変形特性に関する研究

野上 良, 榎 亮, 登尾 啓史*¹

Deformation Properties of Three Kinds of MSD Models of Rheology Object Calibrated by Randomized Algorithm

Ryo Nogami, Ryo Enoki, Hiroshi Noborio*¹

Abstract – In this paper, we propose three kinds of mass-spring-damper (MSD) models of rheology object, and experimentally evaluate the best one concerning to shape and volume accuracies. The MSD model requires a few cost to calculate force propagation and shape deformation of rheology object. For this reason, the dynamic animation can be made by a personal computer within the video-frame rate (about twenty milli-seconds). Moreover, in order to maintain deformation precision, we calibrate all coefficients of dampers and springs under many experimental data by the randomized algorithm. Then in the set of simple pushing operations, volume and shape of virtual rheology object based on the best model are extremely similar to these of real rheology object. This is a case study to generate dynamic animation efficiently and precisely by the MSD model.

Keywords : Rheology Object, Residual Displacement, MSD (Mass-Spring-Damper) model, Uncertain Parameter Calibration, Randomized Algorithm

1. はじめに

残留変位を有するレオロジー物体のモデリングは、まだ未知の分野である。人間がそれを仮想空間で操作するとき、またロボットアームがそれを実空間で操作するとき、このモデリングは不可欠である。それらは、ロボティクス（遠隔操作など）、仮想現実感（ダイナミックアニメーションなど）、医療（仮想手術など）、組み立て作業（食品加工など）、およびゲーム・アミューズメントなどで利用される。レオロジー物体はもちろん、粘弾性体や弾性体の変形過程をダイナミックアニメーションで眺めたり、その力伝播過程をハプティック（触覚フィードバックデバイス）で体感したりするとき、少しの計算時間で正確な力や形状の得られることが望ましい。一般に、その精度と計算時間の間にはトレードオフの関係がある。したがって、変形精度（リアリティ）にこだわらなければ、通常のPCやグラフィックスボードで、インタラクティブな変形操作は十分可能である。しかし、その精度にこだわりながらリアルタイムで体感するには、現在でもスーパーコンピュータや専用ハードウェアに頼らざるを得ない[1]～[3]。

これまで、弾性体や粘弾性体のモデリングでは、マス・バネ・ダンパ法（MSD法）[4]～[8]、差分法（FDM

法）[9]、境界要素法（BEM法）[10],[11]、有限要素法（FEM法）[12]～[15],[17]がよく使われてきた。特に、力伝播や質点移動が正確に取り扱える有限要素法はよく利用されており、この計算時間の（事前に計算できる）ところを前処理するなどして[1],[16]短縮を取り扱った研究は多い。しかし、実際の物体と仮想の物体を比較して、それらの力伝播や形状変化の精度をきちんと比較した研究は少なく、レオロジー物体を取り扱ったものはさらに少数である[18]～[20]。

そこで本論文では、残留変位や戻り変位を直接制御できるマス・バネ・ダンパ法に注目する。一般に、力伝播や形状変化を高速計算できるMSD法の問題は、それらの精度がよくないことである。本研究では、まずレオロジー物体のモデルとして3次元格子構造モデルを採用し、Voigtモデルとダンパが直列につながったものをその基本要素とする。この構造や基本要素はすでに使われており、簡単なパラメータ同定もされている[18]～[20]。同様に、実験データにもとづいてマス、バネ、ダンパの係数を同定するというアイデアも既に提案されているが、本格的な実験やモデルの検討はなされていない[7],[8]。

そこで本研究では、実験データから基本要素のバネとダンパの係数をランダム化アルゴリズムで同定し、力伝播や形状変化の精度がよくないというMSD法の問題点を克服する。ここでは、数種類の押し方でロボットアームがパン生地を押し、押ししている間と離

*1: 大阪電気通信大学大学院 工学研究科

*1: Graduate School of Engineering, Osaka Electro-Communication University

した後の4つの時刻において、実物体と仮想物体の形状の差異を計測し、それを小さくするようにランダム化アルゴリズムで良好な係数の集合を選び出す。

本論文では、まず2章において、弾性や粘性の特性を有する基本要素、およびそれが多数配置された3次元格子構造モデルについて説明する。また、力や変形の計算手法を説明する。さらに、合致しない物体の部分に注目し、最初のMSDモデルから2つの拡張モデルを作成する。3章では、基本要素を構成するバネとダンパの係数の同定方法を説明する。まず、実レオロジー物体と仮想レオロジー物体の形状の違いを定量的に評価する。次に、その評価値にもとづくランダム化アルゴリズムより、仮想レオロジー物体の3係数を同定する。4章では、パラメータ同定が終了した3モデルの優劣を、計算時間、記憶容量、および形状変化の精度の観点から評価する。最後に5章では、まとめや今後の課題を述べる。

2. 3つのMSDモデル

本章では、1つのバネと2つのダンパからなる基本要素、およびそれらが多数配置された3次元格子構造モデルを説明する。そして、そのモデルにおける力伝播や形状変化の計算方法を説明し、基本MSDモデルを拡張した2つのMSDモデルを紹介する。

2.1 Voigtモデルとダンパを備えた要素

本研究では、図1に示すVoigt部とダンパ部(長さの比は1:1)からなる基本要素を利用する。この基本要素は、論文[18]~[20]の基本要素と似ている。論文[18]では、VoigtモデルとMaxcellモデルから成る基本要素を用いている。論文[19],[20]では、Voigtモデルおよび適応性のあるダンパから成る基本要素を用いている。本研究では、適応性のあるダンパを適応性のないダンパにおきかえ、各要素の係数が簡単にキャリブレーションできるようにしている。

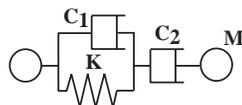


図1 基本要素
Fig.1 Basic element

この基本要素の主な特性を説明する。まず、Voigt部において、 C_1 の適切な値に比べて K の値が小さすぎると、粘性変形のように遅い戻り方となる。一方、 C_1 の適切な値に比べて K の値が大きすぎると、弾性変形のような速い戻り方となる。また、Voigt部とダンパ部の対比において、 K や C_1 の適切な値に比べて C_2 の値が大きすぎると、変位が残留し塑性変形となり、小さすぎると残留変位はなく元の形に戻ってしまう。最後に、係数 K, C_1, C_2 が適切な値なら、レオロジー物体は押されても図2(a)のように安定に変形する。しかし、それらの一つでも限界を超えてしま

と、基本要素の両端の2質点の位置が逆転し、図2(b)のようにレオロジー物体の形状は発散してしまう。

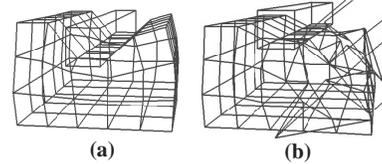


図2 (a) 安定したレオロジー物体の形状
(b) 不安定なレオロジー物体の形状
Fig.2 (a) A stable shape of a rheology object.
(b) An unstable shape of the object.

2.2 格子構造モデル(モデル1)

本研究では、レオロジー物体の様々な変形を表現するため、3次元格子構造モデルを利用する[18]~[20]。ここでは、 X, Y, Z 軸に沿って、等間隔に質点を配置し、レオロジー物体の自然な変形を表現する(図3(a))。ここで、格子の1辺の長さを l とすると、隣接する質点間の長さは $l, l \times \sqrt{2}, l \times \sqrt{3}$ の3種類となる。また、 N を質点数、 M_{object} を物体の質量とすると、それぞれの質点 M は $M = M_{object}/N$ となる。

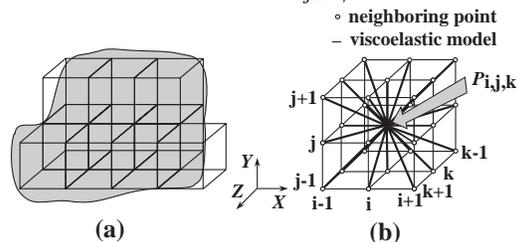


図3 (a) レオロジー物体を表現する3次元格子構造モデル (b) 隣接質点と基本要素の配置
Fig.3 (a) A 3-D voxel/lattice model of rheology object (b) Neighboring mass points connected by the basic elements.

基本要素は、全ての隣接質点間に配置される(図3(b))。レオロジー物体の変形は、各基本要素の変形により表現される。仮想レオロジー物体を1辺の長さが l の $5 \times 3 \times 5$ 個の格子に分割すると、 $6 \times 4 \times 6$ 個の質点が配置される。質点 (i, j, k) ($1 \leq i \leq 6, 1 \leq j \leq 4, 1 \leq k \leq 6$)の位置ベクトルを $P_{i,j,k}$ とする。ここで、質点 $P_{i,j,k}$ の運動方程式を導出する。 α, β, γ をそれぞれ0または ± 1 とすると、質点 $P_{i,j,k}$ の隣接点は $P_{i+\alpha, j+\beta, k+\gamma}$ となる。質点 $P_{i,j,k}$ とその隣接点 $P_{i+\alpha, j+\beta, k+\gamma}$ 間の基本要素によって、質点 $P_{i,j,k}$ に働く力を $F_{i,j,k}^{\alpha, \beta, \gamma}$ で表す。このとき、 $P_{i,j,k}$ に働く内力 $F_{i,j,k}^e$ は、 $F_{i,j,k}^{\alpha, \beta, \gamma}$ の合計で与えられる。

$$F_{i,j,k}^e = \sum_{\substack{\alpha, \beta, \gamma \in \{-1, 0, 1\} \\ (\alpha, \beta, \gamma) \neq (0, 0, 0)}} F_{i,j,k}^{\alpha, \beta, \gamma} \quad (1)$$

ここで、質点 $P_{i,j,k}$ に働くすべての外力の和を $F_{i,j,k}^o$ とすると、次の運動方程式が得られる。

$$M \ddot{P}_{i,j,k} = F_{i,j,k}^e + F_{i,j,k}^o \quad (2)$$

全ての質点 $P_{i,j,k}$ において、位置を得るために運動方程式を解く必要がある。これはRunge-Kutta法で解

いているが、計算時間がかかってしまう。但し、剛体と床に接する質点の位置の計算は不要である。剛体に接する質点は、押し下げられた量だけ移動し、床に接する質点は不動である。このようにするとレオロジー物体の変形シミュレーションが可能である。

最後に、このモデル（基本要素、格子構造モデル、運動方程式など）と類似のものは、既に提案されているが [18] ~ [21]、これだと押し続けている間や押し終わったあと、実物体と仮想物体の形状差が大きくなってしまふ。この違いは、主に次の2箇所において顕著である：(1) 剛体に押されてへこんだ付近；(2) レオロジー物体の側面。この主な理由は、実物体と比較して仮想物体が膨張しなかったためである。この形状の不一致を解決するためには、分割する格子数を増加させる方法、および距離が異なる基本要素の3係数を独立にキャリブレーションする方法などが考えられる。しかし、前者の方法では、仮想レオロジー物体の変形計算に要する時間が指数的に膨大になり、MSD モデルの長所である高速性が失われる。一方、後者の方法では、探索空間が指数的に膨大になり、良質のパラメータが得られなくて形状のリアリティを失うことになる。したがって、今回それらの方法は採用せず、体積要素を導入した修正モデルを考える。

2.3 体積要素を含んだ格子構造モデル (モデル 2)

本節では、各々の格子に体積要素を導入し、レオロジー物体を膨らませる方法を提案する。体積要素とは、各格子の重心に新たな質点を配置し、基本要素の数を増加させ、体積を増加させる方法である (図 4(a),(b))。この手法は、論文 [13] で提案され、弾性の等方性または異方性の制御に用いられた。本論文では、これをレオロジー物体のモデリングに活用する。

まず最初に、最も長い隣接質点間の基本要素を除去する (図 4(a))。このとき、格子の重心である質点から、先程の長さの半分の8つの基本要素を加える (図 4(b))。一般に、これら短い基本要素の係数 C_1, C_2, K は、従来の長い基本要素の係数 C_1, C_2, K とは異なる。本研究では、長さを考慮し、従来のものに $\alpha = 0.5$ をかけた値を用いている。

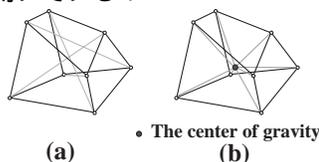


図 4 (a) 対角に位置する質点を結んだ4つの要素 (モデル 1) (b) 格子の重心に新たな質点を配置し、その質点と他の質点を結んだ8つの要素 (モデル 2)

Fig. 4 (a) Four elements directly connecting two opposite vertices in a voxel. (b) Eight elements connecting from the center of gravity of a voxel to its eight vertices.

2.4 特別な内力を考慮した格子構造モデル (モデル 3)

前節で導入した体積要素の影響より、仮想物体が全体的に大きくなる傾向が見られる。これより、外力を加えた付近の上面の形状をあわせようとすると、横面が膨張しすぎてしまう。ここでは、その不一致を直接正すため、横面の形状をあわせるための2種類の特別な内力をモデル 1 に付加する。1つは剛体からの外力が作用する質点到隣接する質点への内力 F_{act} 、もう1つは床からの外力が反作用する質点到隣接する質点への内力 F_{rea} である (図 5(a))。これらは、剛体と床に挟まれたレオロジー物体を外へ強く押し出す直接的な内力である。この内力は最初にのみ加え、それ以降は一切加えない。つまり、レオロジー物体を押し続けたり除重したりする間には、いかなる内力も加えない。したがって、仮想レオロジー物体では内力一定より運動量保存則が成り立つ。本研究では、これ以外の物理法則を仮定しない。

まず、剛体が質点を押す Y 軸方向の外力を考える。このとき、特別な内力 F は点 G から質点へ方向に働く。点 G は以下で説明する線と面の交点である (図 5(b))。線とは、剛体とレオロジー物体の接面の重心を含み、かつ剛体からの外力と同じ方向成分を持つ直線である。面とは、レオロジー物体の重心 G_{object} を含む水平面である。このとき、内力 F は3つの要素 F_x, F_y, F_z に分解される (図 5(c),(d))。一方、垂直方向に床の質点が押されたとき、特別な内力 F は点 G から質点へ方向に働く。このとき、内力 F は2つの要素 F_x, F_z に分解される (図 5(c),(d))。

次に、剛体が質点を押す Y 軸方向以外 (ここでは X 軸方向) の外力を考える。このとき、特別な内力 F は点 G から質点へ方向に働く。このとき、内力 F は3つの要素 F_x, F_y, F_z に分解される (図 5(c),(e))。ここで、X 軸に関して、内力要素の方向を外力要素の方向の逆に設定する (図 5(c)(e) では F_x)。一方、X 軸方向に床の質点が押されたとき、特別な内力 F は点 G から質点へ方向に働く。このとき、内力 F は2つの要素 F_x, F_z に分解される (図 5(c),(e))。ここで、X 軸に関して、内力要素の方向を外力要素の方向の逆に設定する (図 5(c)(e) では F_x)。

剛体からの外力が作用する質点と距離 l で隣接する質点 (図 5(d),(e) では黒く大きい点)、および距離 $\sqrt{2} \times l$ で隣接する質点 (図 5(d),(e) では黒く小さい点) に特別な内力 F が分配される。ここで、これらの内力の合計は、剛体から作用した外力の合計と等しい。これは、各要素毎に大きさだけで計算される。そして、外力の作用点に近い質点により多くの内力を配分するため、その比率を 2 : 1 とした。一方、床からの抗力

が反作用する質点と距離 l で隣接する質点 (図 5(d),(e) では大きい四角形), および距離 $\sqrt{2} \times l$ で隣接する質点 (図 5(d),(e) では小さい四角形) に特別な内力 F が分配される. ここで, これらの内力の合計は, 床から反作用した外力の合計と等しい. そして, 外力の作用点に近い質点により多くの内力を配分するため, その比率を 2 : 1 とした.

2.5 双一次補間を用いた各質点への力の伝達方法

MSD モデルの利点は, 力伝播や変形に要する計算時間が少ないことである. これより, 仮想レオロジー物体の格子数を増加させることは, 計算時間の増加に直接つながり好ましくない. そこで, 剛体とレオロジー物体の間で離散的に力を伝達する方法を提案する.

まず, 剛体の側面を仮想レオロジー物体の格子 (長さ l) と同じ格子で分割する (図 6(a)). これより, 同じ格子とそれより小さい格子 (長さ m, n ($m, n \leq l$)) が生まれる. そして, 強い力がかかる質点 (例えば f_3^*) と, 弱い力がかかる質点 (例えば f_1^*, f_2^*, f_4^*) が生じる. これらの質点の力は, 各々の距離 (m, n) に依存する. 例えば, 図 6(a) では, 強い力がかかる質点は f_3^* のみであり, その他の質点の力は $f_2^* = m/l \times f_3^* (0.25 \times l \leq m)$, $f_4^* = n/l \times f_3^* (0.25 \times l \leq n)$, $f_1^* = (m^2 + n^2)/l^2 \times f_3^* (0.25 \times l \leq m, 0.25 \times l \leq n)$ となる.

一方, レオロジー物体は格子の集合として表現される (図 6(b)). 本研究では, 同じ外力を複数の試行で与えるため, ロボット・マニピュレータの手先に取り付けた剛体でレオロジー物体を変形させる.

剛体の側面で垂直方向に押したとき, レオロジー物体は徐々に変形する. 一般に, 剛体から与えられる力 f^* のほとんどは, レオロジー物体の格子の内部に作用する. このとき, 方向は変わらないが, 力 f^* は 4 つの力 f_a, f_b, f_c, f_d に分配される. それらは 4 つに分割されたレオロジー物体の格子の面積に反比例するように決められる (図 6(c)). たとえば, 図 6(b) の丸で囲まれた部分にあたる図 6(c) では, 外力 $f^* = l^2$ は, $f_a = s \times t$ [area a] ($0.25 \times l \leq s, 0.25 \times l \leq t$), $f_b = (l - s) \times t$ [area b] ($s \leq 0.75 \times l, 0.25 \times l \leq t$), $f_c = (1 - s) \times (l - t)$ [area c] ($s \leq 0.75 \times l, t \leq 0.75 \times l$), $f_d = s \times (1 - t)$ [area d] ($0.25 \times l \leq s, t \leq 0.75 \times l$) の面積の比率で分配される.

3. 修正ランダムイズドアルゴリズムによる係数

K, C_1 および C_2 の測定

本章では, まず実物体と仮想物体の形状の誤差を計測し, 次にそれが最小となる係数 K, C_1, C_2 をランダムイズドアルゴリズムで同定する.

3.1 実物体と仮想物体の比較方法

本研究では, 実際のレオロジー物体として, 小麦粉と水を混ぜたものを使用した. 実物体の大きさは

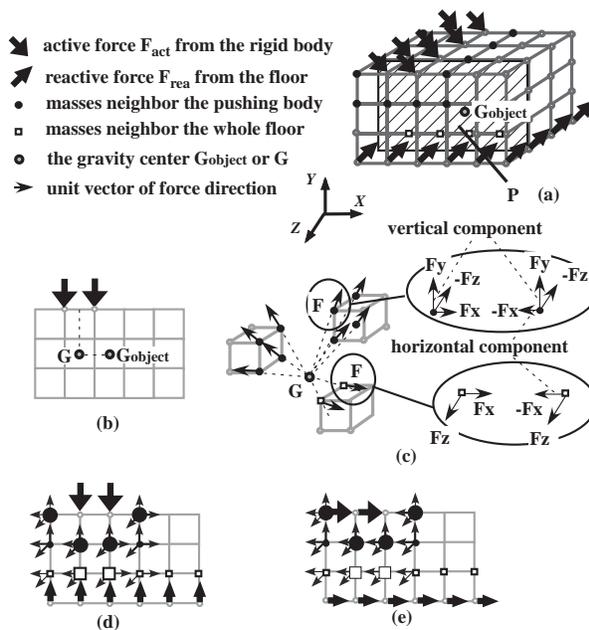


図 5 (a) 3次元レオロジー物体に作用する特別な内力 (b) 点 G は線と面の交点である. 線とは, 剛体がレオロジー物体に接する面の重心を含み, 剛体からの外力と方向成分が同じ直線である. 面とは, レオロジー物体の重心 G_{object} を含む水平面である. (c) 内力 F は点 G から各質点に与えられる. それは要素 F_x, F_y, F_z から構成される. (d)(a) の切断面 P における Y 軸方向の内力生成を説明. 剛体からの外力に対して, 要素 F_x, F_y, F_z の内力, 床の外力から要素 F_x, F_z の内力が分配される. (e)(a) の切断面 P における Y 軸方向以外 (ここでは X 軸方向) の内力生成を説明. 剛体からの外力に対して, 要素 F_x, F_y, F_z の内力, 床の外力から要素 F_x, F_z の内力が分配される. このとき, X 軸方向について, 内力要素 F_x の方向は外力要素のそれと逆に設定される.

Fig. 5 (a) There are two sets of special internal forces and its affected mass points in a 3-D rheology object. (b) The point G is the intersection between line and plane. The line whose direction coincides with the direction of external force includes the center of gravity of encountered area between rheology and rigid objects. On the other hand, the plane is the horizontal one including the center G_{object} of gravity. (c) The internal force F from G to each mass point and its three component forces F_x, F_y and F_z . (d) F_x, F_y and F_z for active external forces and F_x and F_z for reactive external forces on a sliced plane P . (e) F_x, F_y and F_z for active external forces and F_x and F_z for reactive external forces on a sliced plane P . The direction of F_x is opposite to the direction of X-component of active force.

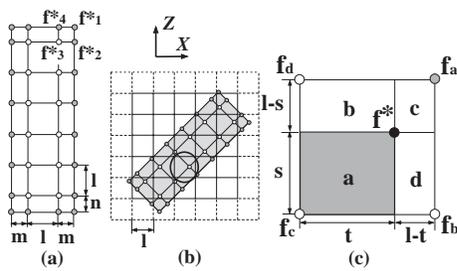


図6 (a) 剛体の側面の格子の集合 (b) レオロジー物体と剛体の格子が接する表面の関係 (c) 力 f^* が加えられたとき、4つの力 f_a, f_b, f_c, f_d の分配方法

Fig. 6 (a) A set of cells around a rigid body. (b) A cell-based relationship between encountered surfaces of a rheology object and a rigid body. (c) An original force f^* is distributed into four forces f_a, f_b, f_c and f_d .

$10[cm] \times 6[cm] \times 10[cm]$ (X,Y,Z軸) とし、仮想物体は実物体の半分の大きさとした ($5[cm] \times 3[cm] \times 5[cm]$)。それゆえ、実物体の体積 ($600[cm^3]$) は仮想物体の体積 ($75[cm^3]$) の8倍となる。本研究では、基本要素のVoigt部とダンパ部の長さの比を固定しているので、基本要素が生み出す力は両端の質点間の距離と速度から一意に定まる。これより、仮想物体の相似則は成立し、仮想物体をX,Y,Z軸方向に等倍率で縮小拡大しても問題は発生しない。最後に、本実験では $M = 6.0$, $M_{object} = 864 [g]$, $N = 144$ を採用した。

ここでは、マニピュレータの手先に取り付けられた剛体で、レオロジー物体は正確に押される (図7(a))。この形状変化は、3台のステレオビジョン・カメラシステム *Digiclops* とそのソフトウェア開発キット (SDK) で計測される。本実験では、前面と左右の横面の形状を測定した。ここで、物体の形状やその押し方より対称性が成り立つので、後面は前面と同形状とした。このカメラ1台で約1000点以上の実物体の表面点群をリアルタイムで取り込める。また、次の原始的な方法で変形後の実レオロジー物体の体積を測る。まず、水でボールを一杯にし、そこに変形した物体を入れる (Fig.7(b))。次に溢れた水を計る。この結果、レオロジー物体の体積は約3%減少していることがわかった。

次に、仮想レオロジー物体は $5 \times 3 \times 5$ 個の6面体で構成される。したがって、*Digiclops* で測定した点から仮想物体 (75個の6面体の集合) までの最小距離を計算するため、Lin-Canny algorithm [22] を用いた。このとき、*Digiclops* の計測誤差の上限は $0.05[cm]$ なので、それ以上の距離誤差のみ加算している。本研究では、押ししている間と除重後の4つの時刻の距離誤差の合計 S で評価する。ランダムイズドアルゴリズムで距離誤差の合計 S を最小化すると、2つの粘性要素と1つの弾性要素に対応する係数 C_1, C_2, K が同定される [23]。

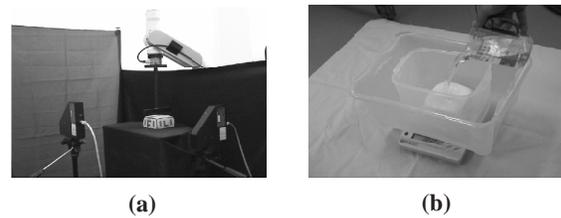


図7 (a) 実験システム: 実レオロジー物体はロボット・マニピュレータの手先に取り付けられた剛体によって押される, そして変形形状は2台の *Digiclops* によって測定される. (b) 水を溢れさせることによって変形前と変形後のレオロジー物体の体積を測定したときの写真

Fig. 7 (a) An experiment system: A real rheology object is pushed by a rectangular rigid body located at the tip of a robotic manipulator, and deformation of its sides is measured by two *Digiclops* cameras. (b) A photo when we measure volume of a rheology object before and after pushing by overflowed water.

3.2 局所最小の係数集合を得るための最急降下法

- 2つのパラメータ T_{cal} および T_{ran} を与える。 T_{cal} とは、ランダムイズドアルゴリズムを終了させる計算時間の閾値である。一方、 T_{ran} とは、探索パラメータを Δ だけ増加・減少させる回数である。
- 探索範囲で係数 K, C_1, C_2 を初期化する。
- 実物体と仮想物体の距離誤差の合計 S を計算する。
- 全ての近傍を見つけるため、 K, C_1, C_2 を Δ だけ増減させる。3つの係数だと8つの近傍が得られる。そして、それぞれの近傍を値 S で評価し、値 S が最小の近傍を得る。
- もし、この新しい最小値 S が現在の最小値 S よりも小さければ、 K, C_1, C_2 を Δ だけ増減させてステップ3にもどる。そうでなければ、現在の最小値 S を得た K, C_1, C_2 を局所最小解として、このアルゴリズムを終了する。

3.3 提案のランダムイズドアルゴリズム

- 3つの係数 K, C_1, C_2 を $K^{min} \leq K \leq K^{max}$, $C_1^{min} \leq C_1 \leq C_1^{max}$, $C_2^{min} \leq C_2 \leq C_2^{max}$ の範囲でランダムに選択する。
- 前述のアルゴリズムを用いて、選択された K, C_1, C_2 から局所最小解を選び出し、 $S_{ran} = S$ とする。そのとき、もし計算時間が T_{cal} よりも大きければ、アルゴリズムを終了し、そうでなければ、ステップ3へ進む。この計算時間とは、全ての処理 (仮想物体の変形、距離誤差の計算、ランダムイズドアルゴリズムなど) に要した時間である。
- 3つの係数 K, C_1, C_2 を、 Δ だけ T_{ran} 回ランダムに増減させる。このとき、選択した K, C_1, C_2 から S を計算し、もし $S \leq S_{ran}$ が成り立てば、ステップ2に戻り、そうでなければ、ステップ3を続ける。

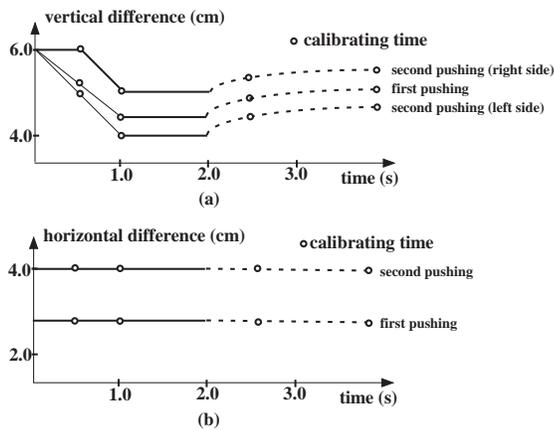


図8 (a)Y軸方向への押し込み距離 (b)X軸方向への押し込み距離

Fig. 8 (a) Pushing distance along the Y-axis direction. (b) Pushing distance along the X-axis direction.

4. MSDモデルの比較結果

本章では、3種類のMSDモデルについて計算時間、記憶容量、そして形状精度を比較する。仮想レオロジー物体の変形を、3次元グラフィックス・アクセラレータボード (NVIDIA Quadro 2EX, 32MB) を搭載したPC (CPU: Pentium4 2.26GHz, Main memory: 1024MB) で計算し、3次元グラフィックスソフトウェア Open GL で描画する。

4.1 計算時間や記憶容量の評価

仮想空間では、実験で測定した4[s]間でレオロジー物体を2000回変形させている。これより、Runge-Kutta法のサンプリングタイムは2[msec]であることがわかる。また、仮想物体の変形シミュレーションには約40[s]要する。これより、1回の変形計算に約20[msec]要することがわかる(表1)。これはビデオレート(33[msec])よりも十分小さいので、リアルなダイナミックアニメーションがオンラインで眺められる(表1)。

第2章で述べたが、モデル1と3の質点数は同じだがモデル1と2の質点数は異なる。各質点の力計算は、複数の力を加算するだけなのでほとんど無視できるが、質点の位置を決定する積分計算は時間がかかる。ここで、力計算は全ての質点に必要なが、積分計算はそうではない。例えば、床に接する質点や剛体に接する質点の位置は、積分計算を用いなくても求められる。また、モデル2においては、格子の重心にある質点に積分計算する必要はない。

前述の理由から、3つのモデルの計算時間は、積分計算が必要な質点数 $N_{cal} = (N_x \times N_y \times N_z) - f_{out} - (N_x \times N_z)$ (N_x, N_y, N_z : X, Y, Z軸の質点数, f_{out} : 押す剛体に接する質点数) に依存する。表1に示すように、押し方1では、 $N_{cal} = (6 \times 4 \times 6) - 12 - (6 \times 6) = 96$ (質点数 $6 \times 4 \times 6$), $182(7 \times 5 \times 7)$, $304(8 \times 6 \times 8)$ となる。そして、各々の計算時間は $N_{cal} = (N_x \times N_y \times$

$N_z) - f_{out} - (N_x \times N_z)$ に比例する。

3種類のモデルの質点数は等しいため記憶容量は同じである。モデル2において、重心の質点は常に重心にあるため、メモリを確保することなくそれを計算できる。記憶容量 m は、質点の記憶容量 m_N (N : 質点数), C++ Compiler (VCC++6.0) といったソフトウェアの記憶容量 m_S を統合したものである。前者は質点数に比例するが、後者は不変である。表1の結果から、 $m_S = 19000[KB]$, $m_N = 16 \times N[KB]$ となった。

表1 PC (CPU: Pentium4 2.26GHz, Main memory: 1024MB) における1回の変形に必要な計算時間 [msec] と記憶容量 [KB]

Table 1 Calculation time [msec] and memory storage [KB] per one deformation in PC (CPU: Pentium4 2.26GHz, Main memory: 1024MB)

Calculation time for the first pushing [msec]			
Number of total masses	[6×4×6]	[7×5×7]	[8×6×8]
(model 1)	18.508	35.125	60.316
(model 2)	19.553	37.141	62.805
(model 3)	18.790	35.860	60.328
Memory storage [KB]			
(model 1)	21416	23044	25260
(model 2)	21452	23048	25360
(model 3)	21492	23044	25315

4.2 レオロジー物体の形状精度の評価

本章では、まずランダムイズドアルゴリズムで良好な3つの係数を選択し、次に仮想物体の変形精度を評価する。そのため、同じアルゴリズム、同じパラメータ $\Delta = 10$, $T_{cal} = 30$ [hour], $T_{ran} = 100$ [number] を用いる。ランダムイズドアルゴリズムの探索範囲は、 $[K^{min}, K^{max}]$, $[C_1^{min}, C_1^{max}]$, $[C_2^{min}, C_2^{max}]$ を $[100, 3000]$, $[500, 10000]$, $[500, 20000]$ とする。この範囲を $\Delta = 10$ で分割すると、537225000点の探索点が誕生する。 $\Delta = 10$ は、実験を重ねた上で最も良好だったので採用した。もし、 $\Delta < 10$ を採用すると、探索点が増加し、良好な点を見つけるのに範囲が大きすぎる。一方、 $\Delta > 10$ を採用すると、探索点が大きすぎて、良好な点を見つけるのに精度が悪すぎる。

ある3係数について、実物体と仮想物体の違いを計算するには、約40[s]の時間を要する。本研究では、3つの係数を計算するのに $T_{cal} = 30$ [hour] を使用したので、探索可能な点は約2700点となった。このことから探索密度が小さいので、 $T_{ran} = 100$ [number] とし、さらに現在の距離誤差 S が、今までの距離誤差 S_{ran} よりも小さければ初期値として採用しないというアルゴリズムを用いた。

レオロジー物体を押したり離したりして、その形状変化の特性を確かめる。本研究では、2種類の押し方を利用する。両方の押し方とも、Y軸方向のみに押しものとする。押し方1では、剛体の右端と左端で同時

に押ししている (図 9) . 実物体と仮想物体の距離誤差や体積精度は表 2 , 除重後の形状変化を図 10 に示す . 表 2 と図 10 が共に示すように , 形状や体積の精度はモデル 3 が最良であった .

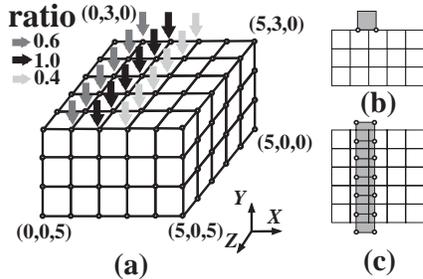


図 9 押し方 1 (a)3次元 (b)前面 (c)上面
Fig. 9 (a) 3-D view for the first pushing. (b) Front view. (c) Upper view.

表 2 押し方 1 での 3 つのモデルの測定結果

Table 2 Calibration results for the first pushing in three models. (*S*: The sum of error distances)

The number of captured points is $N = 15235$						
Calibration result	S [cm]	S/N [cm]	K [gf/cm ³]	$C1$ [gfs/cm ³]	$C2$ [gfs/cm ³]	Volume [cm ³]
(model 1)	2097.40	0.138	1990	510	3470	68.49
(model 2)	2308.62	0.152	1720	1300	780	83.88
(model 3)	1785.78	0.117	3000	1380	1550	69.95

The number of points whose errors are more than 0.25cm					
	first	second	third	fourth	total
(model 1)	513	381	335	267	1496
(model 2)	498	1451	459	388	2796
(model 3)	266	224	242	177	895

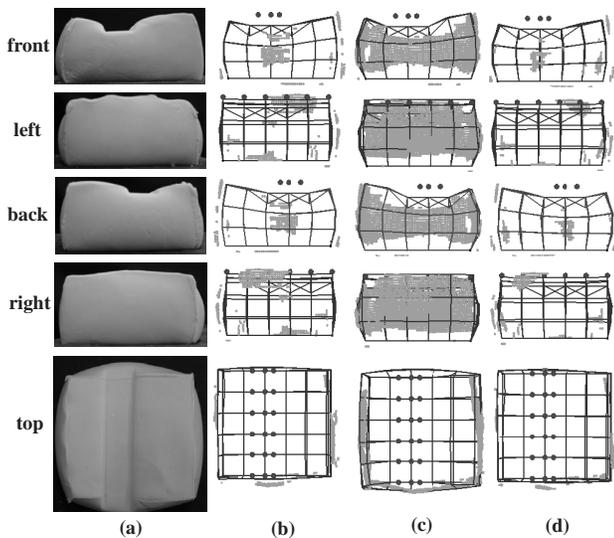


図 10 押し方 1 での実物体と仮想物体の距離誤差 . (a) 実レオロジー物体 . (b) , (c) , (d) 仮想レオロジー物体 (モデル 1,2,3) . 灰色の点は距離誤差が大きいところを示す .

Fig. 10 The distance error between real and virtual rheology objects for the first pushing. (a) Real rheology object. (b),(c),(d) Virtual rheology objects which are colored by gray, whose errors are larger in the proposed models 1, 2 and 3.

押し方 2 は , 剛体の右端と左端で時間差を与えて押

すものである (図 11) . 実物体と仮想物体の距離誤差や体積精度は表 3 , 除重後の形状変化を図 12 に示す . 表 3 と図 12 が共に示すように , 形状や体積の精度は同じくモデル 3 が最良であった .

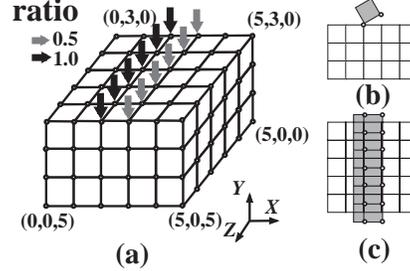


図 11 押し方 2 (a)3次元 (b)前面 (c)上面
Fig. 11 (a) 3-D view for the second pushing. (b) Front view. (c) Upper view.

表 3 押し方 2 での 3 つのモデルの測定結果

Table 3 Calibration results for the second pushing in three models. (*S*: The sum of error distances)

The number of captured points is $N = 14551$						
Calibration result	S [cm]	S/N [cm]	K [gf/cm ³]	$C1$ [gfs/cm ³]	$C2$ [gfs/cm ³]	Volume [cm ³]
(model 1)	2483.15	0.171	1970	510	3470	67.95
(model 2)	2746.35	0.189	1780	980	720	88.95
(model 3)	2051.96	0.141	1950	7520	8930	70.28

The number of points whose errors are more than 0.25cm					
	first	second	third	fourth	total
(model 1)	1260	983	724	751	3718
(model 2)	1002	1629	755	832	4218
(model 3)	837	604	496	485	2422

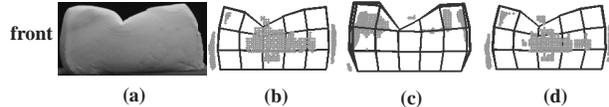


図 12 押し方 2 での実物体と仮想物体の距離誤差 . (a) 実レオロジー物体 . (b) , (c) , (d) 仮想レオロジー物体 (モデル 1,2,3) . 灰色の点は距離誤差が大きいところを示す .

Fig. 12 The distance error between real and virtual rheology objects for the second pushing. (a) Real rheology object. (b),(c),(d) Virtual rheology objects which are colored by gray, whose errors are larger in the proposed models 1, 2 and 3.

5. おわりに

本論文では , 3 種類の MSD モデルで仮想レオロジー物体を表現し , 多数の実験データに基づくランダムイズドアルゴリズムで良好な 3 係数を選定した . 変形中のレオロジー物体の力伝播は , MSD モデルより効率的に計算され , その変形は PC 上で観察できる . さらに , 今回選択した押し方では , 仮想物体の形状や体積は実物体のそれらと近いものになった . 力を加える方向が Y 軸方向で , かつ力がそれほど大きくない場合 , 形状と体積の精度が最良なのはモデル 3 であった .

今後の課題としては , 3 次元 4 面体構造モデル [2] との比較 , および別の基本要素 (例えば , マス・パネ要

素, および Voigt と Maxwell の要素 [18], [20]) でのモデリングなどがある. また, 未知係数のキャリブレーションアルゴリズムとして GA (Generic Algorithm) を採用することも今後の課題である.

謝辞

日頃から有益な議論と情報の提供を頂く平井慎一教授および田中弘美教授 (立命館大学大学院理工学研究科) に感謝を致します. なお, この研究の一部は, 2002 年度 文部科学省科学研究費補助金 (No.14550247), および 2003 年度 大阪電気通信大学 共同研究 (A) の補助を受けています.

参考文献

- [1] Y.Zhuang, "Real-time Simulation of Physically-Realistic Global Deformations," *Doctoral Thesis, UC Berkeley*, 2000.
- [2] G.Debunne, M.Desbrun, M.Paule Cani and A.Barr, "Dynamic Real-time Deformations using Space and Time Adaptive Sampling," *Computer Graphics (Proc. SIGGRAPH)*, pp.31-36, 2001.
- [3] X.Wu, M.S.Downes, T.Goktekin and F.Tendick, "Adaptive Nonlinear Finite Elements for Deformable Body Simulation Using Dynamic Progressive Meshes," *Computer Graphics Forum (Proc. Eurographics)*, pp.349-358, 2001.
- [4] K.Waters, "A Muscle Model for Animating Three-Dimensional Facial Expression," *Computer Graphics (Proc. SIGGRAPH)*, pp.17-24, 1987.
- [5] E.Promayon, P.Bacconnier and C.Puech, "Physically-Based Deformations Constrained in Displacements and Volume," *Computer Graphics Forum (Proc. Eurographics)*, pp.155-164, 1996.
- [6] 田中厚子, 広田光一, 金子豊久, "力覚表現を考慮した仮想物体の変形手法," *情報処理学会論文誌*, Vol.39, No.8, pp.2485-2493, 1998.
- [7] J.Louchet, X.Provot and D.Crochemore, "Evolutionary Identification of Cloth Animation Models," *Proc. of 6th Workshop on Computer Animation and Simulation (EGCAS'95)*, pp 44-54, 1995.
- [8] J.Louchet, M.Boccaro, D.Crochemore and X.Provot, "Building New Tools for Synthetic Image Animation using Evolutionary Techniques," *Proc. of the Artificial Evolution '95*, pp 273-286, 1996.
- [9] D.Terzopoulos, J.Platt, A.Barr and K.Fleischer, "Elastically Deformable Models," *Computer Graphics (Proc. SIGGRAPH)*, pp.205-214, 1987.
- [10] M.B.Nielsen and S.Cotin, "Real-Time Volumetric Deformable Models for Surgery Simulation using Finite Elements and Condensation," *Computer Graphics Forum (Proc. Eurographics)*, pp.57-66, 1996.
- [11] D.James and D.K.Pai, "ArtDefo, Accurate Real Time Deformable Objects," *Computer Graphics (Proc. SIGGRAPH)*, pp.65-72, 1999.
- [12] D.Terzopoulos and K.Fleisher, "Modeling Inelastic Deformation: Viscoelasticity, Plasticity, Fracture," *Computer Graphics (Proc. SIGGRAPH)*, pp.269-278, 1988.
- [13] G.Debunne, M.Desbrun, M.P.Cani, A.Barr, "Adaptive Simulation of Soft Bodies in Real-Time," *Proc. of Computer Animation 2000 (CA'00)*, pp.133-144, 2000.
- [14] M.Muller, J.Dorsey, L.McMillan, "Real-Time Simulation of Deformation and Fracture of Stiff Ma-

terials", *Eurographics Computer Animation and Simulation 2001*, Springer-Verlag Wien, pp.113-124, 2001.

- [15] M.Muller, L.McMillan, J.Dorsey, R.Jagnow and B.Cutler, "Stable Real-Time Deformations", *Proc. of ACM SIGGRAPH Symp. on Computer Animation*, pp 49-54, 2002.
- [16] 広田光一, 金子豊久, "柔らかい仮想物体の力覚表現," *情報処理学会論文誌*, Vol.39, No.12, pp.3261-3268, 1998.
- [17] J.Berkley, "Haptic Rendering of Deformable Bodies using Real-Time Finite Element Analysis: An Application to Suturing Simulation," *Doctoral Thesis, University of Washington*, 2002.
- [18] S.Tokumoto, Y.Fujita and S.Hirai, "Deformation Modeling of Viscoelastic Objects for Their Shape Control," *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp.1050-1057, 1999.
- [19] 徳本真一, 平井慎一, "形状制御のためのレオロジー物体の変形モデリング," *日本ロボット学会誌*, Vol.18, No.8, pp.1141-1148, 2000.
- [20] S.Tokumoto, S.Hirai, and H.Tanaka, "Constructing Virtual Rheological Objects," *Proc. of the World Multiconference on Systemics, Cybernetics and Infomatics*, pp.106-111, 2001.
- [21] H.Noborio, R.Enoki, S.Nishimoto and T.Tanemura, "On the Calibration of Deformation Model of Rheology Object by a Modified Randomized Algorithm," *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2003 (to appear).
- [22] M.C.Lin and J.F.Canny, "A Fast Algorithm for Incremental Distance Calculation," *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp.1008-1014, 1991.
- [23] J.-C.Latombe, "Robot Motion Planning", *Kluwer Academic Publishers.*, 1991.

(2003 年 5 月 7 日受付)

[著者紹介]

野上 良



2003 年大阪電気通信大学総合情報学部情報工学科卒業. 現在, 同大学院工学研究科修士課程情報工学専攻に在学中. 人工現実感に関する研究に従事.

榎 亮



2001 年大阪電気通信大学総合情報学部情報工学科卒業. 2003 年同大学院工学研究科修士課程情報工学専攻修了. 現在, 日本フィッツ (株) において IT に関する実務に従事.

登尾 啓史 (正会員)



1987 年大阪大学大学院基礎工学研究科博士課程修了, 同年大阪大学基礎工学部機械工学科助手, 88 年大阪電気通信大学工学部精密工学科専任講師, 現在同総合情報学部情報工学科教授. ロボット工学, コンピュータビジョン, コンピュータグラフィックス, 人工現実感などの研究に従事 (工学博士).

Precise Deformation of Rheology MSD Model Calibrated by Randomized Algorithm

Hiroshi Noborio, Ryo Nogami and Ryo Enoki

Division of Information and Computer Science
Graduate School of Engineering
Osaka Electro-Communication University
Hatsu-Cho 18-8, Neyagawa, Osaka 572-8530, Japan

Abstract

In this paper, we propose and compare three kinds of mass-spring-damper (MSD) models of rheology object, and experimentally select the best model concerning to shape deformation and volume accuracy. The MSD model requires a few costs to calculate force propagation and shape deformation of rheology object. For this reason, the dynamic animation can be made by a personal computer within the video-frame rate (about twenty milli-seconds). Moreover, in order to maintain deformation precision, we calibrate all coefficients of dampers and springs under many experimental data by the randomized algorithm. Then in the set of simple pushing experimental operations, shape deformation and volume of virtual rheology object based on the best model is similar to these of real rheology object. This is a case study to generate dynamic animation efficiently and precisely by the MSD model.

1. Introduction

Real-time simulation of deformable object is a younger field. Dynamic animation is indispensable in robotics and virtual reality, which has been aggressively used in tele-operation, humanoid, assembly and task planning, computer animation, game and amusement and so on. The key trade-off occurs between calculation time and deformation/propagation accuracy. In general, shape deformation should be calculated in 33 milli seconds for the video-frame rate, and furthermore force propagation should be calculated in a few milli seconds for the haptic rendering. If this trade-off is broken, the animation becomes off-line, otherwise, it becomes on-line. The performance of modern computer and graphics hardware has made physical-based animation possible in real time. But even with today's best hardware and most sophisticated technique^{1,2,3}, only a few hundred elements with small deformations have been simulated in real-time. To represent models of elastic and visco-elastic objects, we have used one of four approaches, i.e., the mass-spring-damper (MSD) method^{4,5,6,7}, the finite difference method (FDM)⁸, the boundary element method (BEM)^{9,10}, and the finite element method (FEM)^{11,12,13,14,15}. The computation efficiency decreases in this order, and the deformation/propagation ac-

curacy increases in this order. To solve such a trade-off between computation efficiency and shape accuracy, almost all the researchers focus on FEM to save computational complexity.

As the opposite major flow, we focus on MSD while maintaining deformation precision in this paper. First of all, we consider a rheology object and its deformation. Since a rheology object always leaves a residual displacement, its model should keep the displacement by many kinds of pushing operations. Although elastic and visco-elastic objects have been aggressively modeled, but a rheology object is seldom modeled^{16,17}. Also, calibrating many kinds of models from a lot of experimental data is not still established^{6,7}.

On the observation, we propose an efficient MSD method for representing flexible deformations precisely. In order to watch shape deformation of rheology object within the video-frame rate (about 20 ms), we adopt the MSD. It is more efficient than the other models. They are typically done off-line, that is, computers spend a few seconds, minutes or hours to arrive at a single answer. The main defective point of MSD for the practical purpose is the lack of deformation accuracy. To overcome this problem, we calibrate three coefficients of each element from many experimental data, which

consists of one spring and two dampers. As the calibration approach, we adopt the randomized algorithm to investigate a set of good coefficients from deformations by many pushing operations^{18,19}.

In this paper, section 2 describes a basic voxel/lattice structure which consists of three elements, i.e., two dampers and one spring. In addition, we explain how to calculate shape deformation and force propagation in the structure. Three elements are dependently used in the dynamic equation. It is represented as the quadratic differential equation. This can be approximately calculated by the fourth-order Runge-Kutta method. Moreover, we propose two extended structures composed of the same elements. Section 3 explains how to calibrate three coefficients of each element. In section 4, we compare three virtual rheology models with the real rheology object pushed by simple operations. First of all, we explain how to evaluate each difference between real and virtual rheology objects. Then, by an efficient randomized algorithm based on the difference, we calibrate three element coefficients so as to construct a virtual rheology object flexibly. Finally in section 5, we will give a few conclusions and future works.

2. Three Kinds of Mass-Spring-Damper Models

In this section, we propose three kinds of mass-spring-damper (MSD) models. First of all, we introduce our element with two dampers and one spring. Then, we explain three kinds of MSD models. The model 1 forms a basic voxel/lattice structure with three lengths of elements. In this model, we explain how to calculate propagation of internal forces in each rheology object. In order to maintain the volume constant condition in the model 2, we exchange four longest elements with eight half-length elements in each voxel in the model 1. In order to deform a few parts whose differences of virtual and real rheology objects are too large in the model 3, we add a set of extra internal forces initially in the model 1. Finally, we propose a digitalized approach to transmit a set of active external forces from a rigid body to a rheology object via their encountered surface. The rheology object is digitalized as a lot of mass points. Therefore, we develop a virtual (digital) force transmission which approximates to a real (analog) force transmission.

2.1. Our Element with Voigt Model and Damper

First of all, we introduce our element which consists of Voigt model and damper serially as shown in Fig.1. This is similar to elements proposed in two researches^{16,17}. In the paper¹⁶, the element is tried, which consists of Voigt and Maxwell models serially. In the paper¹⁷, the element is investigated, which consists of Voigt model and an adaptive damper serially. The adaptive damper flexibly controls coefficient of damper during pushing and releasing operations. In our element, the left Voigt model represents viscosity and elasticity, and the right damper expresses some residual displacement. The former Voigt model generates many kinds of elastic and visco-elastic materials. The latter damper makes many properties of rheology object.

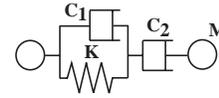


Figure 1: Our basic element with two dampers and one spring.

We briefly introduce what kinds of properties three coefficients generate. The larger the coefficient K is, the stronger the elasticity is. K controls displacement of deformation behavior. Moreover, the larger the coefficient C_1 is, the stronger the viscosity is. C_1 controls speed of the behavior. Finally, the larger the coefficient C_2 is, the larger the residual displacement is. If C_2 is small enough, the object appears elastic or visco-elastic property. On the other hand, if C_2 is large enough, the object appears plastic property. Furthermore, if K, C_1, C_2 are adequately selected within [100,3000], [500,10000], [500,20000], the virtual rheology object pushed freely is stably deformed as Fig.2(a). If some of K, C_1, C_2 are too large, each element becomes unstable and consequently the shape of rheology model is crushed as Fig.2(b). If K, C_1, C_2 are too small, each element loses elasticity, viscosity, and residual displacement, respectively.

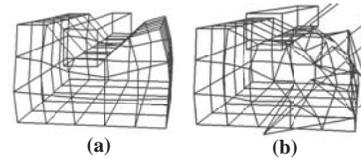


Figure 2: (a) A stable shape of a rheology model. (b) An unstable shape of the model.

2.2. A Basic Voxel/Lattice Model (Model 1)

A rheology object deforms in a 3-D environment. In order to describe several kinds of deformations flexibly, we adopt a symmetric voxel/lattice structure to describe a rheology object^{16,17}. In the structure, let us distribute mass points uniformly in a rheology object whose intervals are the same along $X, Y,$ and Z axes (Fig.3(a)). Let N be the number of mass points and M_{object} be the total mass of rheology object. Therefore, each mass is given by $M = M_{object}/N$.

In our experiment, a real rheology object is made by mixing wheat flour and water. The rheology object (10cm \times 6cm \times 10cm divisions) is horizontally and vertically two times larger than its virtual rheology object (5cm \times 3cm \times 5cm divisions). Here, the unit length l is defined as 1cm in a virtual environment. Therefore, the former volume (600 [cm³] = water weight [g]) is eight times larger than the latter volume (75 [cm³]). In the experiment, we use $M = 6.0$, $M_{object} = 864$ [g], $N = 144$ ($= 6 \times 4 \times 6$).

The elements are inserted between all the neighboring mass points as illustrated in Fig.3(b). The virtual rheology object is deformed by expanding and contracting all the elements. Let $P_{i,j,k}$ be position vector corresponding to mass point (i, j, k) ($1 \leq i \leq 6, 1 \leq j \leq 4$ and $1 \leq k \leq 6$). Let us

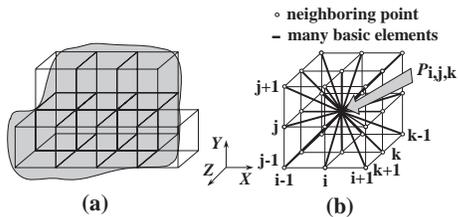


Figure 3: (a) A voxel/lattice model of rheology object (b) Neighboring mass points connecting by many basic elements.

derive quadratic differential equation of each mass at $P_{i,j,k}$. Each internal force acting on $P_{i,j,k}$ by the element between $P_{i,j,k}$ and its neighboring point $P_{i+\alpha,j+\beta,k+\gamma}$ is denoted by $F_{i,j,k}^{\alpha,\beta,\gamma}$. For each mass point, 6 elements whose distance is denoted as l ($=1\text{cm}$), 12 elements whose distance is denoted as $\sqrt{2}l$ ($=\sqrt{2}\text{cm}$), and 8 elements whose distance is denoted as $\sqrt{3}l$ ($=\sqrt{3}\text{cm}$) are located. Therefore, total internal force $F_{i,j,k}^e$ acting on $P_{i,j,k}$ is given by the sum of 26 ($=6 + 12 + 8$) internal forces $F_{i,j,k}^{\alpha,\beta,\gamma}$. Moreover, if the sum of active external forces at $P_{i,j,k}$ is denoted by $F_{i,j,k}^\alpha$, we can obtain the following quadratic differential equation. This summation is not so expensive on calculation time.

$$M\ddot{P}_{i,j,k} = F_{i,j,k}^e + F_{i,j,k}^\alpha \quad (1)$$

In order to calculate next position $P_{i,j,k}$ ($1 \leq i \leq 6$, $1 \leq j \leq 4$ and $1 \leq k \leq 6$) at each mass, we should solve the above differential equation. This is done by the fourth Runge-Kutta method, but it is expensive. If a human operator gives an active external force at a mass point, its next position is calculated. By the expanding and contracting its neighbor elements, adequate internal forces are received at all connecting masses. Then, their next positions are calculated in parallel. This serial or parallel propagation starts from a set of pushed masses. Finally after determining all mass positions, we can describe a pushed virtual object in a 3-D graphics PC.

$$F_{i,j,k}^e = \sum_{\substack{\alpha,\beta,\gamma \in \{-1,0,1\} \\ (\alpha,\beta,\gamma) \neq (0,0,0)}} F_{i,j,k}^{\alpha,\beta,\gamma} \quad (2)$$

We note that mass positions on an encountered surface between a rheology object and its pushing rigid body, and on the whole floor are fixed. This means the integrations by the Runge-Kutta method can be neglected at these points.

Finally in this model and the other models described in previous works^{16,17}, we understand that shape of a calculated (virtual) rheology object unfortunately differs from shape of a practical (real) one. The shape differences mainly occur at two areas: (1) upper side around pushing surface; and (2) four sides of rheology object. This phenomenon occurs because of weak expansion forces. The reason is that volume of an experimental (real) object is always larger than volume which is eight times larger than volume of a calculated (virtual) object. To overcome this drawback, we construct two kinds of models in the following paragraphs.

2.3. Voxel/Lattice Model Including Volume Constant Condition in the Model 1 (Model 2)

In our previous work²⁰, we understand shape of a calculated (virtual) rheology object is not easily much the same as shape of an experimental (real) object. Especially, eight times larger volume of the former object is too small against volume of the latter object after the releasing. To overcome this drawback, we expand volume of each voxel by volume constant condition. The volume constant condition extends a voxel during deformation by eliminating four longest elements (whose distances are to be $\sqrt{3}\text{cm}$ as shown in Fig.4(a)) and adding eight half-length elements from its center of gravity to eight vertices (whose distances are to be $\sqrt{3}/2\text{cm}$ as illustrated in Fig.4(b)).

This technique has been already used in a mass-spring model¹². This controls the isotropy or anisotropy of some elastic material. This idea can be straightforwardly extended to a mass-spring-damper model as this paper. Three coefficient of K , C_1 and C_2 in each of eight shorter elements are defined by multiplying α to K , C_1 and C_2 in each of four longest elements. Note that α is set as 0.5 because each of four elements is two times longer than each of eight elements.

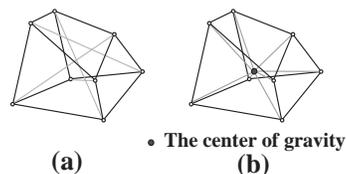


Figure 4: (a) Four longest elements always connect two opposite vertices in a voxel. (b) Eight shorter elements connect from the gravity center of a voxel to its eight vertices.

2.4. Voxel/Lattice Model Adding Extra Internal Forces in the Model 1 (Model 3)

With the support of volume constant condition, eight times larger volume of virtual (calculated) rheology object is similar to that of real (experimental) rheology object. However, this property is not always stable. It depends on the set of coefficients. The virtual object volume is sometimes too large because of stronger diagonal forces within each voxel. Moreover, shape of the former object still differs from that of the latter object. Especially, the swelling of four sides and that beside pushing surface are not enough. To overcome this tendency, we propose the model 3 to add extra internal forces into all mass points neighboring pushed mass points on the rigid body and the floor. The set of internal forces at masses neighboring masses pushed by a rigid body is the same against the set of active external forces received from the rigid body. Also, the set of internal forces neighboring masses on the floor is the same against the set of reactive external forces received from the whole floor.

In this paragraph, we explain how to calculate two kinds of extra internal forces in this model. If a rheology object is pushed by a rigid body, a volume between pushed area

on the upper side and the bottom side on the floor is forced out. To express this phenomenon, we add two sets of special internal forces initially (Fig.5(a)). One is a set of internal forces at mass points neighboring upper mass points pushed by active forces F_{act} from a rigid body, and another is a set of internal forces at mass points neighboring bottom mass points pushed by reactive forces F_{rea} from a floor.

First of all, at each mass point neighboring mass point pushed by a rigid body along the vertical axis, e.g., Y-axis, we consider an extra internal force F whose direction is from G to the mass point. The point G is the intersection between line and plane (Fig.5(b)). The line includes the gravity center of encountered area between rheology and rigid objects, whose direction is coincident with a pushing direction. In this paper's experiments, the direction is restricted along the vertical axis, i.e., Y-axis. On the other hand, the plane is defined as the horizontal plane including the gravity center G_{object} of rheology object, i.e., XZ-plane. Then, we decompose each extra internal force F into three components F_x , F_y and F_z (Fig.5(c),(d)). As contrasted with this, at each mass point neighboring mass point pushed from the floor along the vertical axis, i.e., Y-axis, we consider another special internal force F whose direction is from G to the mass point. Then, we decompose each extra internal force F into F_x and F_z (Fig.5(c),(d)).

Secondly, at each mass point neighboring mass point pushed by a rigid body along one of the other axes, e.g., X-axis and Z-axis, we independently consider an extra internal force F whose direction is from G to the mass point. In addition, we decompose each extra internal force F into F_x , F_y and F_z (Fig.5(c),(e)). Finally, we reverse direction of the F_x along X-axis. As contrasted with this, at each mass point neighboring a mass point pushed from the floor along one of the other axes, e.g., X-axis and Z-axis, we independently consider another extra internal force F from G to the mass point. Moreover, we decompose each special internal force F into F_x and F_z (Fig.5(c),(e)). Finally, we reverse direction of the F_x along X-axis.

All masses neighboring masses on a rigid body are received by larger internal forces, whose distances are the unit length. Also, all masses neighboring masses on the rigid body are received by smaller internal forces, whose distances are $\sqrt{2}$ times larger than the unit length. In Fig.5(d),(e), the former masses are described as larger black circles, and the latter masses are illustrated as smaller black circles. The ratio between larger and smaller forces is 2 : 1. The sum of additional internal forces running at all the masses is the same against the sum of external active forces at masses around the rheology object pushed by the rigid object. Moreover, all masses neighboring masses on a rigid floor are received by larger internal forces, whose distances are the unit length. Also, all masses neighboring masses on the rigid floor are received by smaller internal forces, whose distances are $\sqrt{2}$ times larger than the unit length.

In Fig.5(d),(e), the former masses are described as larger squares, and the latter masses are illustrated as smaller squares. The ratio between larger and smaller forces is 2 : 1. The sum of additional internal forces running at all the masses is the same against the sum of external reactive forces at masses around the rheology object pushed by the rigid floor.

In all trials, two kinds of special internal forces are initially added into the model 1. Therefore, we never add any external force after pushing a rheology object. For this reason, the sum of internal forces in the rheology object is always constant after pushing. As a result, **the conservation law of momentum** is exactly maintained during the deformation.

2.5. A Digital Operation Pushing a Rheology Object by a Rigid Body

The advantage of MSD model is to calculate force propagation and shape deformation efficiently. For this reason, we do not like to divide a virtual rheology object into a lot of voxels. As long as the number of voxels increases, calculation cost and memory storage in three models increase exponentially. To overcome this calculation explosion, we propose a digital operation pushing a rheology object by a rigid body.

In this research, we flexibly push a virtual rheology object by a rigid body, which are digitalized by the unit length l . First of all, surface around a rigid body is uniformly digitalized by the unit length l as illustrated in Fig.6(a). In general, since width and length of the surface cannot be divided exactly, rests m and n appear at ends of the surface ($m, n \leq l$). For this reason, there are major mass points, e.g., f_3^* , and are minor mass points, e.g., f_1^* , f_2^* and f_4^* around the surface. On the other hand, surface around a rheology object is also uniformly digitalized by the unit length l as illustrated in Fig.6(b). In this research, so as to keep a selected force precisely, we push a rheology object by a rigid body located on the tip of a robotic manipulator.

After transmitting many active forces from a digitalized rigid body to another digitalized rheology object, the rheology object starts to deform. In general, each active force f^* around the rigid body usually hits inside a cell around the rheology object. Therefore, a major force f^* should be distributed into four forces f_a , f_b , f_c and f_d at four vertices of a cell around the rheology object (Fig.6(c)). Moreover, magnitudes of four forces f_a , f_b , f_c and f_d are determined as follows: First of all, we define the total force received by each cell as the area $F = l^2$. Secondly, we determine $f_a = s \times t$ [area a] if $0.25 \times l \leq s$ and $0.25 \times l \leq t$, we regard $f_b = (l - s) \times t$ [area b] if $s \leq 0.75 \times l$ and $0.25 \times l \leq t$, we select $f_c = (l - s) \times (l - t)$ [area c] if $s \leq 0.75 \times l$ and $t \leq 0.75 \times l$, and finally we find $f_d = s \times (l - t)$ [area d] if $0.25 \times l \leq s$ and $t \leq 0.75 \times l$. Overall, magnitudes of four forces linearly correspond to their opposite areas. For example, magnitudes of f_a , f_b , f_c and f_d are linearly determined as areas a , b , c and d (Fig.6(c)). Moreover, we should consider a minor force f^*

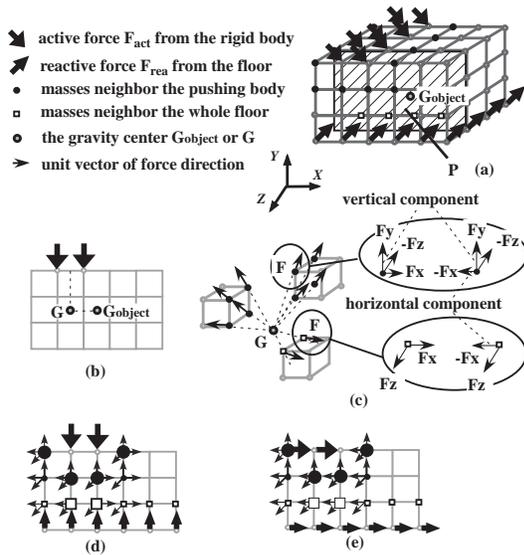


Figure 5: (a) There are two kinds of special internal forces affecting a virtual rheology object. (b) The point G is the intersection between line and plane. The line passes through the gravity center of encountered area between rheology and rigid objects, whose direction equals to the pushing direction. In this research, the line is limited in the vertical axis, i.e., Y -axis. On the other hand, the plane is defined as XZ horizontal plane including the gravity center G_{object} of rheology object. (c) An extra internal force F generated by an active external force consists of three components F_x , F_y and F_z . As contrasted with this, an extra internal force F generated by a reactive external force consists of two components F_x and F_z . (d) F_x , F_y and F_z at masses denoted as black circles are made for active external and vertical forces by a rigid body on an arbitrary sliced plane P . The magnitude of forces at larger black circles is twice bigger than that of forces at smaller black circles. On the other hand, F_x and F_z at masses denoted as squares are made for reactive external and vertical forces by a whole floor on an arbitrary sliced plane P . The magnitude of forces at larger squares is twice bigger than that of forces at smaller squares. (e) F_x , F_y and F_z at masses denoted as black circles are made for active external and horizontal forces by a rigid body on an arbitrary sliced plane P . F_x is to be the opposite force of X -component of the active force. The magnitude of forces at larger black circles is twice bigger than that of forces at smaller black circles. On the other hand, F_x and F_z at masses denoted as squares are made for reactive external and horizontal forces by a whole floor on an arbitrary sliced plane P . F_x is to be the opposite force of X -component of the active force. The magnitude of forces at larger squares is twice bigger than that of forces at smaller squares.

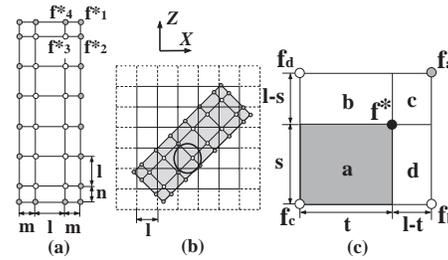


Figure 6: (a) A set of cells around a rigid body. (b) A cell-based relationship between encountered surfaces of a rheology object and a rigid body. (c) An original force f^* is distributed into four forces f_a , f_b , f_c and f_d .

located in the margin of pushing area. That is, we consider a horizontal force $f_h^* = m/l \times f^*$ if $0.25 \times l \leq m$, e.g., f_2^* , we consider a vertical force $f_v^* = n/l \times f^*$ if $0.25 \times l \leq n$, e.g., f_4^* , and we regard a diagonal force $f_d^* = (m^2 + n^2)/l^2 \times f^*$ if $0.25 \times l \leq m$ and $0.25 \times l \leq n$, e.g., f_1^* (Fig.6(a)). This approach is an approximated transformation based on the balance of all analog forces and their distances via a digital encountered surface.

3. Modified Randomized Algorithm for Calibrating C1, C2 and K

The defective point of MSD is accuracy of force propagation and shape deformation. To overcome this, we calibrate K , C_1 and C_2 by minimizing the difference between shape deformations of real and virtual rheology objects in our modified randomized algorithm.

3.1. How to Calculate Shape Difference Between Real and Virtual Rheology Objects

In this research, a rheology object is precisely pushed by a rigid body located at the tip of a robotic manipulator (Fig.7(a)). The deformation, that is, the sequence of shapes is measured by two stereo vision camera systems *Digiclops* and its software development kit (SDK) *Triclops* (provided by Point Grey Research Inc, Canada). Each captures about three or more thousand points as shape of real rheology object in the real-time manner. After capturing the shape deformation, we finally measure how much total volume is changed before pushing and after releasing the rheology object. For this purpose, we use the following primitive method. First of all, we fill a ball with water, and then drop a deformed object into the ball (Fig.7(b)). Secondly, we gather overflowed water and measure its weight by a precise electric balance. As a result, we can understand volume of rheology object always decreases by about three percentages.

In order to evaluate a difference between real and virtual rheology objects, we summarize minimum distances from captured points to their nearest surfaces around a virtual rheology object. A virtual object consists of $5 \times 3 \times 5$ hexahedrons which are individually deformed from initial cubes. Therefore in order to evaluate the difference, we calculate

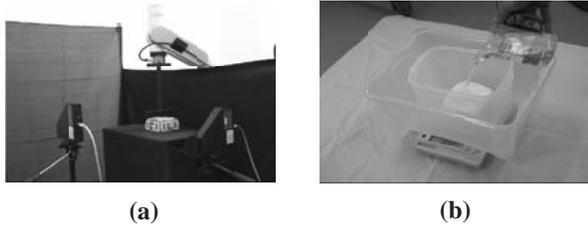


Figure 7: (a) An experiment system: A real rheology object is pushed by a rectangular rigid body located at the tip of a robotic manipulator, and deformation of its sides is measured by two Digiclops cameras. (b) A photo when we measure volume of a rheology object before pushing and after releasing by overflowed water.

the minimum of $5 \times 3 \times 5$ shortest distances for each captured point to all hexahedrons by Lin-Canny closest point algorithm¹⁸, and then we calculate the sum S of all the minimum distances for all captured points, which are smaller than the average error 0.05cm of Digiclops. In our calibration, we use the sum of four S at four times during pushing and after releasing. By minimizing the total sum in an efficient randomized algorithm, we can obtain a better set of three coefficients K , C_1 and C_2 of two dampers and one spring.

3.2. A Steepest Descendent Method

1. Two parameters T_{cal} (threshold of calculation time) and T_{ran} (driving distance of random walk) are given in advance.
2. Initialize coefficients K , C_1 and C_2 in a 3-D search space.
3. We calculate shape difference S between real and virtual rheology objects.
4. In order to find all the possible neighbors, we decrease and increase K , C_1 and C_2 by Δ . In this 3-D case, we obtain eight possibilities, that is, $(K+\Delta, C_1+\Delta, C_2+\Delta)$, $(K+\Delta, C_1+\Delta, C_2-\Delta)$, $(K+\Delta, C_1-\Delta, C_2+\Delta)$, $(K+\Delta, C_1-\Delta, C_2-\Delta)$, $(K-\Delta, C_1+\Delta, C_2+\Delta)$, $(K-\Delta, C_1+\Delta, C_2-\Delta)$, $(K-\Delta, C_1-\Delta, C_2+\Delta)$ and $(K-\Delta, C_1-\Delta, C_2-\Delta)$. Then, after calculating all sums at all the neighbors, we select their minimum.
5. If the minimum is smaller than S obtained in step 3, we move to the neighbor with the minimum by decreasing or increasing K , C_1 and C_2 by Δ , and return to step 3. Otherwise, the algorithm finishes.

3.3. Our Randomized Algorithm

1. We select arbitrary K , C_1 and C_2 within a given 3-D search space, whose ranges are $K^{min} \leq K \leq K^{max}$, $C_1^{min} \leq C_1 \leq C_1^{max}$ and $C_2^{min} \leq C_2 \leq C_2^{max}$.
2. We calculate the S for the K , C_1 and C_2 . By the steepest descendent method described above, we get one of the local minima whose value is the smallest S and set $S_{ran} = S$. Then, if calculation time equals to or is larger than T_{cal} , the algorithm ends, otherwise, move to step 3.
3. We randomly increase and decrease three coefficients K , C_1 and C_2 T_{ran} times by Δ . Then, we calculate S for K , C_1 and C_2 . Then, if $S \leq S_{ran}$ is satisfied, we return to step 2, otherwise, continued to step 3.

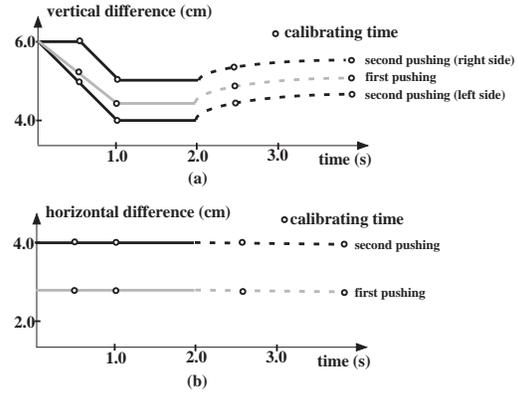


Figure 8: Two pushing operations are described as the gray and black lines. The vertical/horizontal differences of pushing and releasing are described as the whole and dot lines. (a) Y-axis direction. (b) X-axis direction.

4. Comparative Results

In this section, we compare three MSD models with each other concerning to computation time, memory storage and shape accuracy. The deformation of virtual rheology object is calculated and visualized by a 3-D graphics software OpenGL in a personal computer (CPU: Pentium4 2.26GHz, Main memory: 1024MB) with a 3-D graphics acceleration board (NVDPIA Quadro 2EX, 32MB).

4.1. Computation Complexity

As mentioned previously, a real rheology object pushed and released by a rigid body deforms during 4 seconds in all the experiments. On the other hand, 2000 deformations of a virtual rheology object under the models 1, 2, and 3 are calculated during about 40 seconds in the CG environment. As shown in Table 1, we check calculation time less than 20 milli-seconds per one deformation. The speed is enough to make dynamic animation because it is smaller than the video-frame rate, i.e., 33 milli-seconds. Furthermore, we show memory storage of the models 1, 2 and 3 is too small and also relatively constant (Table 1).

The numbers of masses and elements in the models 1 and 3 are the same, but numbers of masses and elements in the models 1 and 2 differ from each other. As mentioned previously, force calculation at each mass is not expensive, but position calculation at the mass is time consuming because of solving the differential equation. The force calculation is always necessary in all the elements, but the position calculation is not always necessary in all the masses. For example, since position of each mass on the floor and the pushing body is fixed and can be calculated without the integration. Moreover in the model 2, since the mass position within each voxel is calculated as the gravity center of the voxel, the integration is not necessary.

For this reason, calculation costs of three models totally depend on the number of mass points $N_{cal} = (N_x \times N_y \times N_z) - f_{out} - (N_x \times N_z)$ (N_x , N_y , N_z : the numbers of masses

along X, Y, Z axes, respectively, f_{out} : the number of masses pushed by a rigid body). As shown in Table 1, since $f_{out} = 12$ is in the first pushing under the $6 \times 4 \times 6$ model, the number of masses to need the integration is denoted as $N_{cal} = (6 \times 4 \times 6) - 12 - (6 \times 6) = 96$. Similarly, the number of masses to need the integration is denoted as 182 and 304 for $7 \times 5 \times 7$ and $8 \times 6 \times 8$ models, respectively. As described in Table 1, calculation time of each model is proportionally evaluated by the equation $N_{cal} = (N_x \times N_y \times N_z) - f_{out} - (N_x \times N_z)$.

Since numbers of mass points in three kinds of models are almost the same, memory storage is also the same. The memory storage m is approximately evaluated by summing the mass storage m_N (N : the number of masses) and basic software storage m_S including C++ compiler (VC++ 6.0). The former is proportional to the number of masses, but the latter is invariable. Using the results in Table 1, we calculate $m_S = 19000[KB]$ and $m_N = 16 \times N[KB]$.

Table 1: Calculation time [msec] per one deformation and memory storage [KB] in PC (CPU: Pentium4 2.26GHz, Main memory: 1024MB)

Calculation time for the first pushing [msec]			
Number of total masses	$6 \times 4 \times 6$	$7 \times 5 \times 7$	$8 \times 6 \times 8$
(model1)	18.508	35.125	60.316
(model2)	19.553	37.141	62.805
(model3)	18.790	35.860	60.328
Memory storage [KB]			
(model1)	21416	23044	25260
(model2)	21452	23048	25360
(model3)	21492	23044	25315

4.2. Deformation Accuracy

In this paragraph, we describe a global aspect of a rheology object by changing three coefficients. In the same randomized algorithm, we use the same parameters $\Delta = 10$, $T_{cal} = 30$ [hour], and $T_{ran} = 100$ [number]. The search space consists of three intervals $[K^{min}, K^{max}]$, $[C_1^{min}, C_1^{max}]$ and $[C_2^{min}, C_2^{max}]$ which are defined by [100,3000], [500,10000] and [500,20000]. If each interval is divided by $\Delta = 10$, the search space includes candidate points whose number is 537225000. The resolution $\Delta = 10$ is experimentally selected in order to find the optimal or a near-optimal solution. If $\Delta < 10$, the number of candidate points is too large to find the solution. If $\Delta > 10$, the magnitude of voxel resolution is too large to find it.

To calculate the sum of differences between real and virtual rheology objects at each candidate point, we need about 40 seconds. In this research, we use $T_{cal} = 30$ [hour] to investigate a better set of three coefficients, and therefore we can check candidate points whose number is about 2700. The search density is too sparse and therefore the randomized algorithm selects a near-optimal solution by using $T_{ran} = 100$ [number] and eliminating candidate points whose sum S is larger than the present minimum sum S_{ran} .

In order to ascertain goodness of pushing a rheology object by a rigid body, we prepare two kinds of pushing. In both

operations, external forces at masses around a rheology object act along the Y-axis. In the first operation, all masses around a rheology object are simultaneously pushed by a rigid body (Fig.9). In this case, we compare which model is the best. Shape and volume differences between real and virtual rheology objects are given in Table 2, and also shapes of both objects after releasing are described in Fig.10. As shown in Table 2 and Fig.10, the model 3 is the best concerning to shape accuracy and volume consistency.

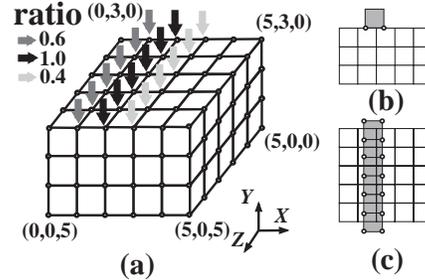


Figure 9: (a) 3-D view for the first pushing. (b) Front view. (c) Upper view.

Table 2: Calibration results for the first pushing in three models. (S : The sum of error distances)

The number of captured points is $N = 15235$						
Calibration result	S [cm]	S/N [cm]	K [gf/cm ³]	$C1$ [gfs/cm ³]	$C2$ [gfs/cm ³]	Volume [cm ³]
(model 1)	2097.40	0.138	1990	510	3470	68.49
(model 2)	2308.62	0.152	1720	1300	780	83.88
(model 3)	1785.78	0.117	3000	1380	1550	69.95
The number of points whose error is more than 0.25cm						
	first	second	third	fourth	total	
(model 1)	513	381	335	267	1496	
(model 2)	498	451	459	388	2796	
(model 3)	266	224	242	177	895	

In the second operation, right and left endpoints of a rigid body contact a rheology object at different times (Fig.11). In this case, shape and volume differences between real and virtual rheology objects are given in Table 3, and also shapes of both objects after releasing are described in Fig.12. As shown in Table 3 and Fig.12, the model 3 is the best concerning to shape accuracy and volume consistency.

Table 3: Calibration results for the second pushing in three models. (S : The sum of error distances)

The number of captured points is $N = 14551$						
Calibration result	S [cm]	S/N [cm]	K [gf/cm ³]	$C1$ [gfs/cm ³]	$C2$ [gfs/cm ³]	Volume [cm ³]
(model 1)	2483.15	0.171	1970	510	3470	67.95
(model 2)	2746.35	0.189	1780	980	720	88.95
(model 3)	2051.96	0.141	1950	7520	8930	70.28
The number of points whose error is more than 0.25cm						
	first	second	third	fourth	total	
(model 1)	1260	983	724	751	3718	
(model 2)	1002	1629	755	832	4218	
(model 3)	837	604	496	485	2422	

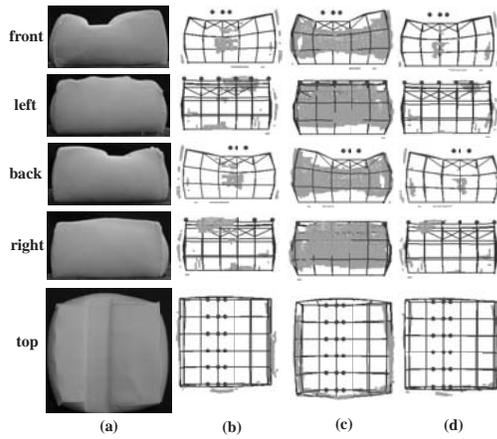


Figure 10: The distance error between real and virtual rheology objects for the first pushing. (a) Real rheology object. (b),(c),(d) Virtual rheology objects which are colored by gray, whose errors are larger in the proposed models 1, 2 and 3.

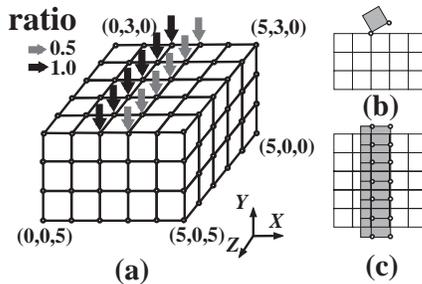


Figure 11: (a) 3-D view for the second pushing. (b) Front view. (c) Upper view.

5. Conclusions and Future Works

In this paper, we represent a rheology object by three kinds of MSD models, and calibrate three coefficients of two dampers and one spring by an efficient randomized algorithm based on many experimental results. Since force propagation of rheology object during each deformation is efficient in the MSD model, we can watch many deformations in the video-frame rate by a personal computer with a popular 3-D graphics acceleration board. Moreover, by the careful calibration as pre-processing, shape and volume of virtual (calculated) rheology object are quite similar to those of real (experimental) one during and after simple pushing operations. As a result, the model 3 is the best concerning to shape and volume accuracy if and only if force directions acting at masses are along the vertical axis, i.e., Y-axis, and also force magnitudes are not so large.

Finally as several future works, another structure (e.g., nested or non-nested tetrahedral meshes²⁾ and another element (e.g. mass-spring element, Voigt and Maxwell element^{16,17)} should be tested by our experimental calibration. In addition, as the calibration algorithm, we should try to use another optimal algorithm such as GA (generic algorithm).

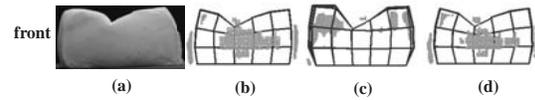


Figure 12: The distance error between real and virtual rheology objects for the second pushing. (a) Real rheology object. (b),(c),(d) Virtual rheology objects which are colored by gray, whose errors are larger in the proposed models 1, 2 and 3.

Acknowledgements

The authors thank Professor S.Hirai and H.Tanaka (Ritsumeikan University) for fruitful discussions. This research is supported in part by 2002 Grants-in-aid for Scientific Research from the Ministry of Education, Science and Culture, Japan (No.14550247).

References

1. Y.Zhuang, "Real-time simulation of physically-realistic global deformations," *Doctoral Thesis, UC Berkeley*, 2000.
2. G.Debunne, M.Desbrun, M.Paule Cani and A.Barr, "Dynamic real-time deformations using space and time adaptive sampling," *Computer Graphics (Proc. SIGGRAPH)*, pp.31-36, 2001.
3. X.Wu, M.S.Downes, T.Goktekin and F.Tendick, "Adaptive nonlinear finite elements for deformable body simulation using dynamic progressive meshes", *Computer Graphics Forum (Proc. Eurographics)*, pp.349-358, 2001.
4. K.Waters, "A muscle model for animating three-dimensional facial expression," *Computer Graphics (Proc. SHIGGRAPH)*, pp.17-24, 1987.
5. E.Promayon, P.Baconnier and C.Puech, "Physically-based deformations constrained in displacements and volume," *Computer Graphics Forum (Proc. Eurographics)*, pp.155-164, 1996.
6. J.Louchet, X.Provot and D.Crochemore, "Evolutionary identification of cloth animation models", *Computer Animation and Simulation (Eurographics)*, pp 44-54, 1995.
7. J.Louchet, M.Boccaro, D.Crochemore and X.Provot, "Building new tools for synthetic image animation using evolutionary techniques", *Artificial Evolution '95 (Springer Verlag: Eds. J.M.Alliot, E.Lutton, E.Ronald, M.Schoenauer and D.Snyers)*, pp 273-286, 1996.
8. D.Terzopoulos, J.Platt, A.Barr and K.Fleischer, "Elastically deformable models," *Computer Graphics (Proc. SIGGRAPH)*, pp.205-214, 1987.
9. M.B.Nielsen and S.Cotin, "Real-time volumetric deformable models for surgery simulation using finite elements and condensation," *Computer Graphics Forum (Proc. Eurographics)*, pp.57-66, 1996.
10. D.James and D.K.Pai, "ArtDefo, accurate real time deformable objects," *Computer Graphics (Proc. SIGGRAPH)*, pp.65-72, 1999.
11. D.Terzopoulos and K.Fleischer, "Modeling inelastic deformation: viscoelasticity, plasticity, fracture," *Computer Graphics (Proc. SIGGRAPH)*, pp.269-278, 1988.
12. G.Debunne, M.Desbrun, M.-P.Cani, A.Barr, "Adaptive simulation of soft bodies in real-time," *Computer Animation 2000*, pp.133-144, 2000.
13. M.Muller, J.Dorsey, L.McMillan, "Real-time simulation of deformation and fracture of stiff materials", *Proceedings of Eurographics, Computer Animation and Simulation (CAS)*, pp.113-124, 2001.
14. M.Muller, J.Dorsey, L.McMillan, R.Jagnow and B.Cutler, "Stable real-time deformations", *Proceedings of ACM SIGGRAPH Symposium on Computer Animation (SCA)*, pp 49-54, 2002.
15. J.J.Berkley, "Haptic rendering of deformable bodies using real-time finite element analysis: an application to surgical simulation," *Doctoral Thesis, University of Washington*, 2002.
16. S.Tokumoto, Y.Fujita and S.Hirai, "Deformation modeling of viscoelastic objects for their shape control," *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp.1050-1057, 1999.
17. S.Tokumoto, S.Hirai and H.Tanaka, "Constructing virtual rheological objects," *Proc. World Multiconference on Systemics, Cybernetics and Informatics*, pp.106-111, July, Auland, 2001.
18. M.C.Lin and J.F.Canny, "A fast algorithm for incremental distance calculation," *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp.1008-1014, 1991.
19. J.-C.Latombe, "Robot motion planning", *Kluwer Academic Publishers.*, 1991.
20. H.Noborio, R.Enoki, S.Nishimoto and T.Tanemura, "On the calibration of deformation model of rheology object by a modified randomized algorithm," *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2003 (to appear).

On the Calibration of Deformation Model of Rheology Object by A Modified Randomized Algorithm

Hiroshi Noborio, Ryo Enoki, Shohei Nishimoto and Takumi Tanemura

Department of Engineering Informatics
Osaka Electro-Communication University
Hatsu-Cho 18-8, Neyagawa, Osaka 572-8530, Japan
nobori@noblab.osakac.ac.jp

Abstract

There are many kinds of rheology objects in our living life. If such rheology objects are individually and completely modeled, we can flexibly deal with such various objects in a factory by a robotic manipulator controlling six degrees-of-freedom forces and moments and also operate them at a 3-D graphics world in a house by a human via a wonderful haptic device feeling the forces and moments. If such a system is developed after acquiring a rheology dynamic model, we can enjoy a clay work in a 3-D virtual environment.

For this purpose, we calibrate deformation model of a rheology object by a modified randomized algorithm based on experimental data. The data is measured from a bread material pushed by a robotic manipulator exactly. The model is a 3-D voxel/lattice structure with many elements, and each element consists of two dampers and one spring. By changing three coefficients of dampers and spring, we can describe various material properties concerning to viscosity and elasticity. In this paper, by minimizing shape difference between real and virtual rheology objects in the near-optimal algorithm, we find a better set of coefficients. Using the calibrated model, we can feel three degrees-of-freedom forces and three degrees-of-freedom moments attracted from a calibrated virtual object in a haptic device and synchronously we can watch shape deformation of the object in a 3-D graphics animation.

1 Introduction

Dynamic animation is indispensable in robotics and virtual reality, which is aggressively used in teleoperation, humanoid, assembly, task planning, game, amusement and so on. The animation is quickly made in PC (e.g., Pentium 4. 2.26GHz) with a graphics accelerator (e.g., NVIDIA Quadro 2EX, 32MB for OpenGL). They are powerful and cheap for making a sequence of 30 full color images. Even though their

motions are extremely complicated, the sequence dynamically describes object movements per the second. From this background, we can easily develop a software to make a graphics animation whose quality of image is high concerning to rendering such as lighting, shading, texture mapping and so on. As an example of this, we use a 3-D graphics software OpenGL in a personal computer (CPU: Pentium4 2.26GHz, Main memory: 1024MB) with a 3-D graphics acceleration board (NVIDIA Quadro 2EX, 32MB) in order to illustrate such a wonderful 3-D graphics animation quickly.

However, it is unfortunately difficult for us to generate exact trajectories of moving objects. The reason is that every object is always affected by complex physical properties in our living space. Some properties are known, but the others are approximated or unknown. For this reason, modeling and calibrating many kinds of physical properties are important in the dynamic animation. In general, the animation includes two behaviors: contact and non-contact behaviors. If two rigid bodies collide with each other, we can calculate a sequence of contact behaviors based on Coulomb and Hertz models. Concerning to this, researchers have proposed many kinds of models for making friction force or impulse and contact force or impulse [1],[2],[3],[4],[5],[6]. If many rigid bodies are connected each other, we can calculate a sequence of non-contact behaviors based on the Newton-Euler equations. To calibrate its dynamic parameters, we use two kinds of approaches using a few special motions and a huge number of general motions, respectively [7],[8],[9].

As contrasted with these works, modeling and calibrating a rheology object is quite backward. This work started before the decade [10],[11],[12], but has not been well developed. Several kinds of elastic objects have been modeled and animated in computer animation or virtual reality. However, some forget many kinds of viscosities, and the others could not be calibrated experimentally [13],[14],[15],[16]. As an exception, a 2-D pixel/lattice structure was calibrated under few experimental results, whose com-

ponent consists of Voigt model (parallel damper and spring) and Maxwell model (serial damper and spring) [17], and then its similar 3-D voxel/lattice structure was calibrated under few experimental results, whose component consists of Voigt model (parallel damper and spring) and one adaptive damper [18]. The models are unfortunately calibrated under few data, and especially coefficients of dampers and springs are calibrated individually. In general, they are strongly depending on each other so as to produce many kinds of material properties. By changing a set of four or three coefficients flexibly, viscosity and elasticity of a rheology object totally increases or decreases. They synchronously appear by changing a set of four or three coefficients. For this reason, we adopt the similar 3-D voxel/lattice structure whose element consists of two dampers and one non-adaptive spring. Then, we calibrate three coefficients synchronously in our efficient randomized algorithm under many kinds of experimental data.

For this purpose, we firstly measure a sequence of deformed shapes several times while pushing a rheology object by a robotic manipulator and synchronously observing it by two stereo vision systems *Digiclops*. Secondly, we minimize a sequence of differences of deformed shapes between virtual and real rheology objects in the calibration. Finally, we watch deformation of the rheology object in a 3-D graphics animation by *OpenGL* and feel its three reactive forces and three reactive moments by six-degrees-of-freedom robot arm *Joyarm*. The system is useful in several practical areas. For example, modeling and calibrating an arbitrary real rheology object can be used in robot assembly [19],[20], the virtual system is frequently used at surgery simulations in medical engineering [21],[21],[22]. Also, they are the basic technique in many application areas [23],[24],[25],[26].

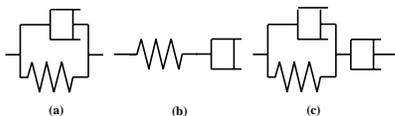


Figure 1 : Visco-elastic elements: (a) Voigt model. (b) Maxwell model. (c) An element with two dampers and one spring

In this paper, section 2 describes Voigt, Maxwell models, and then our element. In addition, we build a 3-D voxel/lattice structure for representing shape deformations and force propagations. Section 3 explains how to calibrate three coefficients of two dampers and one spring in our 3-D structure. The set of coefficients is used in dynamic equation represented as quadratic differential equation. This can be approximately calculated by the Runge-Kutta method. In section 4, we describe two experimental results for a same visco-elastic object by two kinds of pushing. First of all, we explain how to evaluate shape difference between real and virtual visco-elastic object. Secondly, by an efficient randomized algorithm using

a sequence of shape differences, we calibrate three coefficients in order to construct a virtual rheology object accurately. Finally, we will give a few conclusions and future problems in section 5.

2 Model of Visco-Elastic Object

In this section, we will briefly explain a classic model, which products many material properties of some visco-elastic object. As shown in Fig.1(a),(b),(c), famous Voigt and Maxwell models and our component have been used for representing a visco-elastic object. Voigt model consists of spring and damper, which connects neighbor mass points in parallel. On the other hand, Maxwell model is a sequence of spring and damper between neighbor mass points. For example, the mixture is adopted for expressing a rheology object [17]. In this paper, we use an element which consists of Voigt model and one damper.

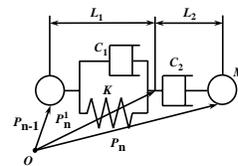


Figure 2 : Our element consists of Voigt model and a damper serially.

2.1 Our mass-spring-damper element

In Fig.2, we formulate some behaviors of our element. First of all, let O be the origin of coordinate system. Let P_{n-1} and P_n be coordinates of element endpoints. Spring and damper coefficients in the Voigt part are denoted as K and C_1 , respectively. The other damper coefficient in the element is denoted as C_2 . The natural length of Voigt part is given by L_1 . Let M be mass at each endpoint. Let P_m be position of connecting point between Voigt and damper parts. Furthermore, $d_n = P_n - P_{n-1}$ is defined. Since these positions P_{n-1} , P_m and P_n exist on a straight line, P_m can be defined by a parameter k as follows: $P_m = kd_n + P_{n-1}$. Here, time varying direction vector is defined as $e_n = d_n / |d_n|$, and also time varying length coefficient is defined as $Z_n = k |d_n|$.

Let F_e be a force applied to a mass point P_n by the classic model. The force F_e equals to a force acting in the Voigt part. Thus, we have the following equation.

$$F_e = -C_1 \dot{Z}_n e_n - K(Z_n - L_1)e_n \quad (1)$$

Also, the force F_e coincides the force acting to a damper part. In consequence, we obtain the following equation.

$$F_e = -C_2 \left(\frac{d}{dt} (|d_n| - Z_n) \right) e_n \quad (2)$$

Here, a force applied to a mass point P_n is defined as F_a . Consequently, the dynamic equation of the mass point P_n is denoted as

$$M\ddot{P}_n = F_e + F_a \quad (3)$$

From three equations (1), (2) and (3), we calculate the dynamic equation of three element model. First of all, by eliminating F_e in the equations (1) and (2), we directly obtain the parameter k . By substituting k into each of equations (1) and (2), we can obtain the value of vector F_e and consequently generate a motion of mass point P_n . The purpose of this research is to select a better set of three coefficients for at least two pushing a given visco-elastic object in a 3-D environment by two kinds of randomized algorithms.

In this paper, while changing values of three coefficients randomly in reasonable ranges by an efficient randomized algorithm, we are seeking for the best set. When the pushing is strong or weak, the best set of three coefficients a little bit differs from each other. However, as shown in our experimental results, the difference is not so large. For this reason, we use one best set for all pushing. Moreover, the classic model generates a wonderful sequence of object shapes, which is totally affected by values of three coefficients.

2.2 3-D voxel/lattice structure

A visco-elastic object deforms in a 3-D environment. In order to describe several kinds of deformations flexibly, we adopt a symmetric 3-D voxel/lattice structure. For this purpose, let us distribute mass points in a natural shape of a visco-elastic object at the same intervals along x, y, and z axes (Fig.3(a)). Let N be the number of mass points and M_{object} be total mass of the object. Therefore, each mass point is given by $M = M_{object}/N$.

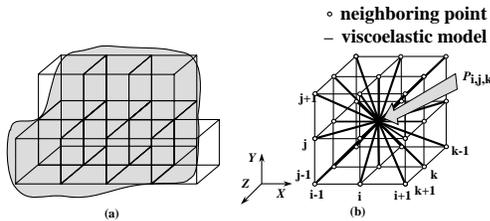


Figure 3 : (a) A 3-D voxel/lattice structure of a visco-elastic object (b) Neighboring lattice mass points.

Our elements are inserted between all neighboring mass points as illustrated in Fig.3(b). Namely, the elements are arranged, whose distances are 1, $\sqrt{2}$, and $\sqrt{3}$ (The unit is the distance between horizontal and vertical neighbor mass points). Visco-elastic

deformation of an object can be represented by deformation of all the elements. Let $P_{i,j,k}$ be position vector corresponding to mass point (i, j, k) . Let us derive motion equation of a mass point at $P_{i,j,k}$. Force acting on $P_{i,j,k}$ by the element between $P_{i,j,k}$ and its neighbor point $P_{i+\alpha, j+\beta, k+\gamma}$ is denoted by $F_{i,j,k}^{\alpha, \beta, \gamma}$. Then, total internal force acting on $P_{i,j,k}$ is given by the sum of $F_{i,j,k}^{\alpha, \beta, \gamma}$, that is,

$$F_{i,j,k}^e = \sum_{\substack{\alpha, \beta, \gamma \in \{-1, 0, 1\} \\ (\alpha, \beta, \gamma) \neq (0, 0, 0)}} F_{i,j,k}^{\alpha, \beta, \gamma} \quad (4)$$

The force $F_{i,j,k}^{\alpha, \beta, \gamma}$ can be computed using a procedure explained in paragraph 2.1. Thus, force $F_{i,j,k}^e$ can be computed by summing all forces. Let $F_{i,j,k}^a$ be a total external force acting on $P_{i,j,k}$. Thus, the equation of motion is described as follows:

$$M\ddot{P}_{i,j,k} = F_{i,j,k}^e + F_{i,j,k}^a \quad (5)$$

By solving a set of equations corresponding to all mass points consisting of the model, we can compute deformation of a visco-elastic object. By calculating successively forces among neighbor mass points, we can obtain forces on all mass points of the above 3-D voxel/lattice structure. Each force between neighbor mass points, position and velocity of each mass point are calculated by the quadratic differential equation. This can be done by the Runge-Kutta method. By these techniques, we can simulate deformation of a pushed visco-elastic object virtually in a 3-D graphics environment.

A 3-D voxel/lattice structure is as follows: The voxel structure consists of $6 \times 4 \times 6$ mass points (Fig.4). All elements are inserted between all neighboring mass points. Therefore, there are $5 \times 3 \times 5$ elements whose distances are 1, and there are $6 \times 4 \times 2$ elements whose distances are $\sqrt{2}$. Moreover, we add the lattice structure into the voxel structure. Therefore, there are $5 \times 3 \times 5 \times 4$ elements whose distances are $\sqrt{3}$.

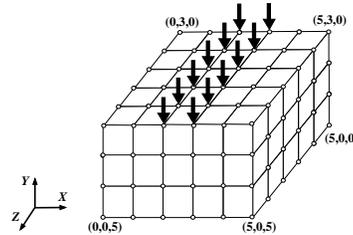


Figure 4 : 12 upper mass points of a rheology object are synchronously pushed by a rectangular object located on the tip of a robotic manipulator.

Let us compute deformation of the model when 12 upper mass points of a rheology object are synchronously pushed by a rectangular object located on the tip of a robotic manipulator (Fig.4). The sum of all mass points is really measured by $M = 6.0$ under $M_{object} = 840$ [g] and $N = 144$ [points].

Also, three coefficients $K[gf/cm^3]$, $C_1[gfs/cm^3]$ and $C_2[gfs/cm^3]$ are experimentally initialized as 400, 2000 and 2000, respectively, in two types of pushing. Finally, we should note that the bottom of the object is fixed to the space. This means 36 bottom mass points of a rheology object are received by the same repulsive forces of calculated attractive forces from its floor.

3 Two Kinds of Randomized Algorithms for Calibrating K, C1 and C2

In the last section, we construct a basic model for representing relations of forces, velocities and positions of many mass points in a rheology object. The model always needs two coefficients C_1 and C_2 of different damper parts and one coefficient K of a spring part. Changing the set of coefficients means changing material of the rheology object. Unfortunately, a rheology object has many aspects depending on absolute magnitude and time difference of a given outer force. This is an interesting property of the rheology object, which differs from rigid, plastic and elastic objects.

In this section, we calibrate K , C_1 and C_2 by minimizing shape differences between real and virtual rheology objects in two kinds of randomized algorithms. Finally, we should note that we did not use force differences acted from real and virtual rheology objects because force information is quite noisy (its magnitude and orientation include 10 and more percent errors).

3.1 How to calculate shape difference between real and virtual rheology objects



Figure 5 : An experiment system: A real rheology object is pushed by a rectangular object located at the tip of a robotic manipulator, and side deformations are simultaneously measured by two *Digiclops* cameras.

In this research, a rheology object is made by mixing wheat flour and water (Fig.6(a)). The scale of the object is denoted as $10\text{cm} \times 6\text{cm} \times 10\text{cm}$. The rheology object is pushed by a rectangular object located at the tip of a robotic manipulator (Fig.5). The deformation, that is, the sequence of shapes of

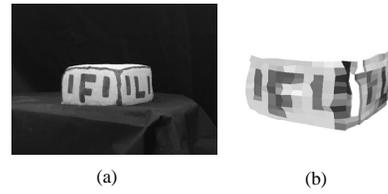


Figure 6 : (a) A real rheology object is built by mixing wheat flour, food red and water. (b) Its virtual rheology object is shown by a point-based computer graphics under about three or more thousand surface points measured by two *Digiclops* cameras.

the real rheology object is measured by the support of two stereo vision camera systems *Digiclops* and its SDK (Software Development Kit) *Triclops* (provided by Point Grey Research Inc, Canada). Each provides real-time 3-D digital image for capturing shapes of the object. A set of about three or more thousand points is captured as shape of the object (Fig.6(b)). Each image has 240×320 pixels with 24bit full color. The set is obtained three times per one second. An average error for capturing our rheology object is about 0.05cm if the distance from *Digiclops* to a rheology object is about 60cm. This error decreases experimentally by changing surface texture of the object and location of the camera and lighting without highlight and shadow. Especially, surface texture is artificially made by mixing food red into our rheology object. Then because of the latter reason, two sides of the rheology object are focused and measured by two stereo vision camera systems *Digiclops*.

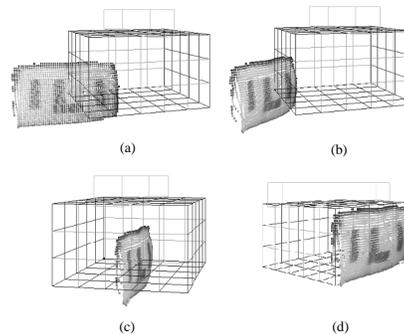


Figure 7 : The real and virtual coordinate systems are coincident with each other by matching their vertices and color landmarks.

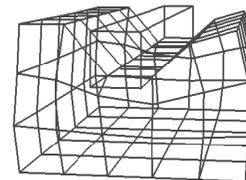


Figure 8 : A simulation system in PC: A virtual rheology object is pushed by a rectangular object, and deformations of five sides are calculated by a quadratic differential equation in our 3-D voxel/lattice structure.

Then, we calculate the sum of minimum distances from a captured point to the nearest surface of a virtual rheology object as follows: First of all, a virtual

object (its coordinate system) is coincident with a real object (its coordinate system) by using their corners and landmarks as illustrated in Fig.7. A virtual object consists of $5 \times 3 \times 5$ hexahedrons individually deformed from cubes (Fig.8). Each hexahedron has six patches which are classified into real and virtual patches. A real patch is always outside a virtual rheology object, and a virtual patch is always inside it. We firstly determine whether a captured point is inside each hexahedron or not. On the assumption that each hexahedron is convex shape, if a captured point is always located in the opposite side of the normal vector of each patch of a hexahedron, the point is inside the hexahedron. In this case, the nearest point is always on one of real patches (is never on edges or vertexes) (Fig.9(a)). Therefore, we only calculate the minimum of shortest distances against real patches. On the other hand, if the point is outside all hexahedrons, we should calculate the minimum of $5 \times 3 \times 5$ shortest distances for all hexahedrons by Lin-Canny closest features algorithm [27] (Fig.9(b)).

Then, after neglecting minimum distances smaller than the average error $0.05cm$ of *Digiclops*, we summarize the other minimum distances from about three or more thousand captured points to their nearest surface of the rheology object. In our calibration, we use the sum S for four sets of captured points during and after each pushing. By minimizing the total S in an efficient randomized algorithm, we can obtain a better set of three coefficients K , C_1 and C_2 .

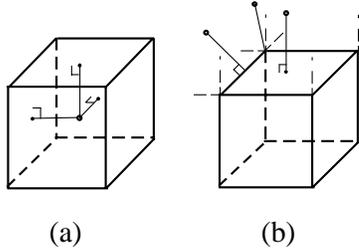


Figure 9 : (a) If a captured point is inside a hexahedron, we calculate the minimum distance from the point to six patches around the hexahedron. (b) Otherwise, we calculate the minimum distance from a captured point to all hexahedrons.

In this section, we optimize K , C_1 and C_2 by using a steepest descent method and two kinds of randomized algorithms. The global (former) randomized algorithm gets local minima from initial points uniformly selected in a given 3-D search space. On the other hand, the local (latter) randomized algorithm finally picks up a better local minimum around the best of all local minima selected in the former [28].

3.2 A Steepest Descent Method

1. Two parameters T_{cal} and T_{ran} are given in advance.
2. Initialize coefficients in a 3-D search space.

3. We calculate shape difference S between real and virtual rheology objects.

4. In order to find all the possible neighbors, we decrease and increase K , C_1 and C_2 by Δ . In this 3-D case, we obtain eight possibilities, that is, $(K+\Delta, C_1+\Delta, C_2+\Delta)$, $(K+\Delta, C_1+\Delta, C_2-\Delta)$, $(K+\Delta, C_1-\Delta, C_2+\Delta)$, $(K+\Delta, C_1-\Delta, C_2-\Delta)$, $(K-\Delta, C_1+\Delta, C_2+\Delta)$, $(K-\Delta, C_1+\Delta, C_2-\Delta)$, $(K-\Delta, C_1-\Delta, C_2+\Delta)$ and $(K-\Delta, C_1-\Delta, C_2-\Delta)$. Then, using S for each neighbor, we select the minimum of sums at all the possible neighbors.

5. If the minimum is smaller than S obtained in step 3, we move to the neighbor with the minimum by decreasing or increasing K , C_1 and C_2 by Δ , and return to step 3. Otherwise, the algorithm finishes.

3.3 A Local Randomized Algorithm

1. By the steepest descent method described above, we get one of the local minima. Then, if calculation time equals to or is smaller than T_{cal} , the algorithm ends with the smallest S , otherwise, move to step 2.

2. We randomly increase and decrease three coefficients K , C_1 and C_2 T_{ran} times by Δ . Then, return to step 1.

3.4 A Global Randomized Algorithm

1. Up to a time threshold T_{cal} , we randomly select initial points within a space whose ranges are $K^{min} \leq K \leq K^{max}$, $C_1^{min} \leq C_1 \leq C_1^{max}$ and $C_2^{min} \leq C_2 \leq C_2^{max}$ in 3-D search space. The density of the initial points is always uniform.

2. For all initial points, we get local or global minima by the steepest descent method described above. Then, the algorithm ends with the smallest S of all the minima.

3.5 An Efficient Algorithm

1. By a global randomized algorithm, we globally find a better local minimum in a given 3-D search space.

2. By a local randomized algorithm from an initial point selected by step 1, we locally find a better local minimum near the initial point [28].

4 Comparative Results

In this section, we calibrate three coefficients of a virtual rheology object during deformations by two types of pushing a real rheology object, which are illustrated in Fig.10. The difference is only the velocity of pushing (direction and orientation are the same). In this research, our rheology object consists of wheat flour and water. During and after

each pushing, deformation of the virtual object is visualized by a 3-D graphics software OpenGL in a personal computer (CPU: Pentium4 2.26GHz, Main memory: 1024MB) with a 3-D graphics acceleration board (NVIDIA Quadro 2EX, 32MB). Also in our virtual reality system, 3-D repulsive forces and 3-D repulsive moments of a rectangular object from a pushed virtual object can be felt by a *Joyarm* (Fig.11).

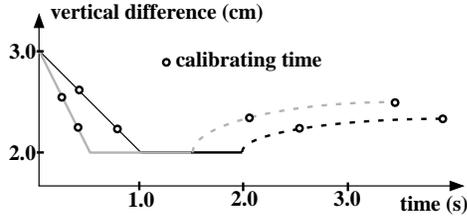


Figure 10 : Weak and strong pushing a rheology object. Weak (first) pushing is described by whole and dotted black line, and strong (second) pushing is shown by whole and dotted gray line.



Figure 11 : A human operator watches deformation of a rheology object by a 3-D graphics software *OpenGL* and feels its three forces and three moments by a six-degrees-of-freedom robotic arm *Joyarm*.

4.1 General Properties

In this paragraph, we describe a global aspect of a rheology object by changing three coefficients. As shown in Fig.12(a), as long as each of K , C_1 and C_2 increases, shape difference S decreases. However, if $C_1 + C_2$ increases extremely, a virtual rheology object cannot converge to a reasonable shape whose neighbor positions of masses are frequently reversed (Fig.12(b)). On the observation, we select search space $[K^{min}, K^{max}], [C_1^{min}, C_1^{max}], [C_2^{min}, C_2^{max}]$ limited by $[100, 2000], [1000, 8000], [5000, 20000]$, respectively.

4.2 The First Pushing

In our efficient 3-D randomized algorithm, we set $\Delta = 1$, $T_{cal} = 30$ [hour], $T_{ran} = 1000$ [number]. After the calibration, we finally obtain coefficients $K[gf/cm^3]$, $C_1[gfs/cm^3]$ and $C_2[gfs/cm^3]$ as 1831, 4732 and 18524.

In the figures 13 and 14, we describe shape transformation of initial and calibrated virtual objects against the real object, respectively. A set of dark gray areas means shape differences are larger than 0.25cm, and another set of light gray areas means

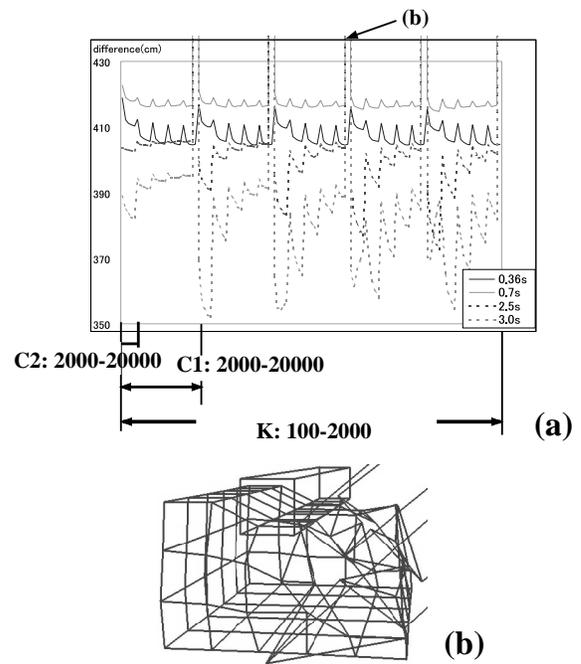


Figure 12 : (a) Shape difference S changes when K , C_1 and C_2 synchronously change within reasonable available ranges. (b) If $C_1 + C_2$ increases extremely, our rheology model does not converge to a unique shape.

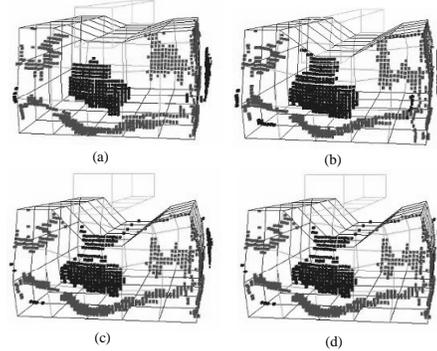


Figure 13 : (a),(b),(c),(d) Four shape differences between real and virtual objects by the first pushing, whose coefficients are initially given as reasonable values $K = 400$ and $C_1 = C_2 = 2000$. A set of dark gray areas means shape differences are larger than 0.25cm, and another set of light gray areas means the differences are less than 0.05cm.

the differences are less than 0.05cm. Therefore, we can see that differences between a calibrated object and its real object are smaller than differences between a non-calibrated object and the real object.

Also in the figure 15, we describe the distribution of differences of minimum distances from about three or more thousand captured points to initial and calibrated virtual objects. In the figure 15(a),(b),(c),(d), numbers of captured points are 34, 14, 90, 101, whose differences are less than 0.1cm, on the other hand, numbers of captured points are -47, -22, -33, 4, whose differences are more than 0.25cm, respectively. From these results, we can see that almost all numbers increase in relatively small differences, on the other hand, almost all numbers decrease in relatively large differences. This means a calibrated object is better than its non-calibrated object against the real

object. As a result, the calibration leads a virtual rheology object that has high viscosities and elasticity. Moreover, by the comparison between calibrated and initial virtual rheology objects with the real rheology object, a calibrated rheology object looks like the real rheology object, and therefore our calibration is meaningful.

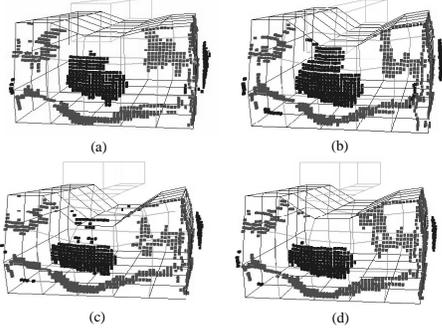


Figure 14 : (a),(b),(c),(d) Four shape differences between real and virtual objects by the first pushing, whose coefficients are completely calibrated as $K = 1831$, $C_1 = 4732$ and $C_2 = 18524$.

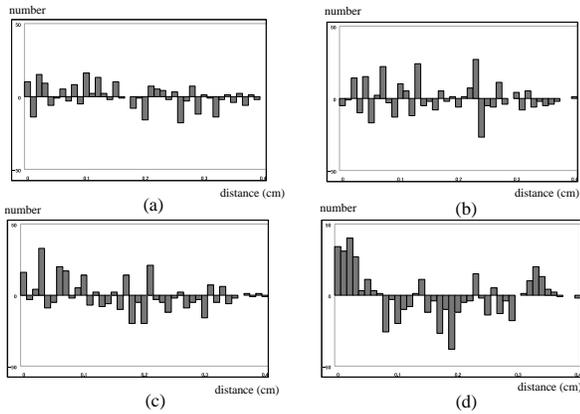


Figure 15 : (a),(b),(c),(d) (the number of captured points at each distance between a calibrated virtual object and its real object) - (the number of captured points at each distance between an initial virtual object and its real object) at four sampling times pushing the objects firstly. In each figure, plus numbers appear in relatively small distances (differences), on the other hand, minus numbers appear in relatively large distances (differences). This means a calibrated object is better than its non-calibrated object against the real object.

4.3 The Second Pushing

In the same algorithm, we use same parameters $\Delta = 1$, $T_{cal} = 30$ [hour], $T_{ran} = 1000$ [number], and finally we obtain calibrated coefficients K [gf/cm³], C_1 [gfs/cm³] and C_2 [gfs/cm³] as 1398, 6345 and 18524.

In the figures 16 and 17, we compare shape transformation of initial and calibrated virtual objects against the real object, respectively. A set of dark gray areas means shape differences are larger than 0.25cm, and another set of light gray areas means the differences are less than 0.05cm. Therefore, we can see that differences between a calibrated object and its real object are smaller than differences between a non-calibrated object and the real object.

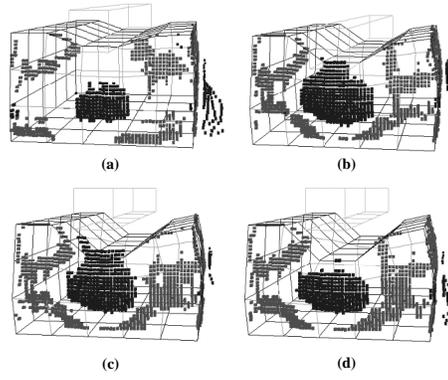


Figure 16 : (a),(b),(c),(d) Four shape differences between real and virtual objects by the second pushing, whose coefficients are initially given as reasonable values $K = 400$ and $C_1 = C_2 = 2000$.

Also in the figure 18, we compare the distribution of differences of minimum distances from about three or more thousand captured points to initial and calibrated virtual objects. In the figure 18(a),(b),(c),(d), numbers of captured points are 87, 40, 35, 107, whose differences are less than 0.1cm, on the other hand, numbers of captured points are -40, -47, 21, -58, whose differences are more than 0.25cm, respectively. From these results, we can see that almost all numbers increase in relatively small differences, on the other hand, almost all numbers decrease in relatively large differences. This means a calibrated object is better than its non-calibrated object against the real object. As a result, a calibrated rheology object is similar to the real rheology object, and therefore our calibration is meaningful.

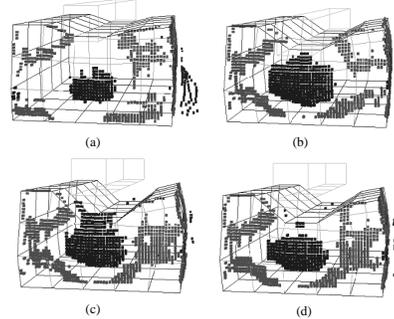


Figure 17 : (a),(b),(c),(d) Four shape differences between real and virtual objects by the second pushing, whose coefficients are completely calibrated as $K = 1398$, $C_1 = 6345$ and $C_2 = 18524$.

5 Conclusions

In this paper, we select a classic 3-D voxel/lattice structure with many mass-damper-spring components as a visco-elastic object, and then calibrate three coefficients of two damper and one spring in each component by an efficient randomized algorithm based on many experimental results. By these approaches, we can watch many deformations by a 3-D graphics animation (*OpenGL*) and feed many forces/moments by a haptics (*Joyarm*). In future, we try to investigate the deformation of rheology object by many kinds of pushing (i.e., translation movements along

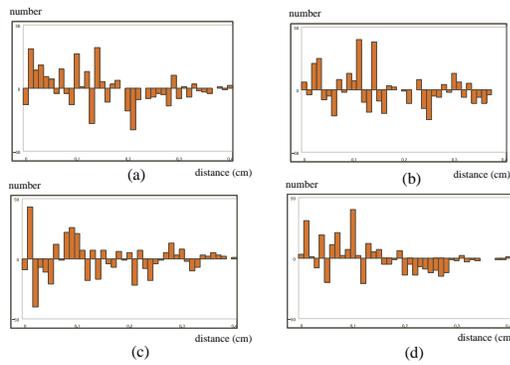


Figure 18 : (a),(b),(c),(d) (the number of captured points at each distance between a calibrated virtual object and its real object) - (the number of captured points at each distance between an initial virtual object and its real object) at four sampling times under pushing the objects secondly. In each figure, plus numbers appear in relatively small distances (differences), on the other hand, minus numbers appear in relatively large distances (differences). This means a calibrated object is better than its non-calibrated object against the real object.

X, Y and Z axes, and also rotation movements centered at X, Y and Z axes, and their combinations). Moreover, another structure (e.g., nested or non-nested tetrahedral meshes) and another component (e.g. mass-spring component, Voigt and Maxwell component) should be tested by our calibration. Finally, as the calibration algorithm, we should try to use another optimal algorithm such as GA (generic algorithm).

Acknowledgments

The authors thank Professor S.Hirai and H.Tanaka (Ritsumeikan University) for the fruitful discussion. This research is supported in part by 2002 Grants-in-aid for Scientific Research from the Ministry of Education, Science and Culture, Japan (No.14550247).

References

- [1] D.Baraff, "Analytical Models for Dynamic Simulation of Non-penetrating Rigid Bodies," *Computer Graphics (Proc. SIGGRAPH)*, Vol.23, pp.223-232, 1989.
- [2] D.Baraff, "Issues in Computing Contact Forces for Non-Penetrating Rigid Bodies," *Algorithmica*, Vol.10, pp.292-352, 1993.
- [3] D.Baraff, "Fast Contact Force Computation Non-Penetrating Rigid Bodies," *Computer Graphics (Proc. SIGGRAPH)*, Vol.28, pp.23-34, 1994.
- [4] B.V.Mirtich, "Impulse-Based Dynamic Simulation of Rigid Body Systems," *Ph.D Dissertation*, University of California at Berkeley, 1996.
- [5] R.Enoki and H.Noborio, "A Comparative Study of Many Randomized Algorithms to Calibrate Virtual Contact and Friction Force Models," *Proc. of the IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, pp.3042-3047, 2002.
- [6] T.Iguchi, N.Katsuyama, H.Noborio and S.Hirai, "Computing and Calibrating Collision Impulses and its Application for Air Hockey Game," *Proc. of the IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, pp.2219-2226, 2002.
- [7] H.Mayeda, K.Osuka and A.Kangawa, "A New Identification Method for Serial Manipulator Arms," *Proc. of the IFAC World Congress*, pp.74-79, 1984.

- [8] H.Kawasaki and K.Nishimura, "Terminal-Link Parameter Estimation of Robotic Manipulators," *IEEE Journal of Robotics and Automation*, pp.485-490, 1988.
- [9] T.Otsuki, T.Iguchi, Y.Murata and H.Noborio, "On the Identification of Robot Parameters by the Classic Calibration Algorithms and Error Absorbing Trees," *Proc. of the IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, pp.1916-1923, 2002.
- [10] A.H.Barr, "Global and local deformations of solid primitives," *Computer Graphics (Proc. SIGGRAPH)*, Vol.18, pp.21-30, 1984.
- [11] T.Sederberg and S.Parry, "Free-Form Deformation of Solid Gemetic Models," *Computer Graphics (Proc. SIGGRAPH)*, Vol.20, pp.151-160, 1986.
- [12] D.Terzopoulos and K.Fleisher, "Modeling Inelastic Deformation: Viscoelasticity, Plasticity, Fracture," *Computer Graphics (Proc. SIGGRAPH)*, Vol.22, No.4, pp.269-278, 1988.
- [13] A.Joukhaider, A.Deguet and C.Laugie, "A Collision Model for Rigid and Deformable Bodies," *Proc. IEEE Int. Conf. on Robotics and Automation*, pp.982-988, 1998.
- [14] Y.Chai and G.R.Luecke, "Virtual Clay Modeling Using the ISU Exoskeleton," *Proc. IEEE Virtual Reality Annual International Symposium*, pp.76-80, 1998.
- [15] G.DeBunne, M.Desbrun, M.P.Cani and A.Barr, "Adaptive Simulation of Soft Bodies in Real-Time," *Computer Animation 2000*, pp.133-144, 2000.
- [16] G.DeBunne, M.Desbrun, M.P.Cani and A.Barr, "Dynamic Real-Time Deformations using Space and Time Adaptive Sampling," *Computer Graphics (Proc. SIGGRAPH)*, pp.31-36, 2001.
- [17] S.Tokumoto, Y.Fujita and S.Hirai, "Deformation Modeling of Viscoelastic Objects for Their Shape Control," *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp.1050-1057, 1999.
- [18] S.Tokumoto, S.Hirai and H.Tanaka, "Constructing Virtual Rheological Objects," *Proc. World Multiconference on Systems, Cybernetics and Informatics*, pp.106-111, 2001.
- [19] Y.F.Zheng, R.Pei and C.Chen, "Strategies for Automatic Assembly of Deformable Objects," *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp.2598-2603, 1991.
- [20] T.Wada, S.Hirai and S.Kawamura, "Indirect Simultaneous Positioning Operations of Extensionally Deformable Objects," *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp.1333-1338, 1998.
- [21] S.Cotin, H.Delingette and N.Ayache, "Real-Time Elastic Deformations of Soft Tissues for Surgery Simulation," *IEEE Transactions on Visualization and Computer Graphics*, Vol.5, pp.62-73, 1999.
- [22] S.P.DiMaio and S.E.Salcudean, "Needle Insertion Modelling and Simulation," *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp.2098-2105, 2002.
- [23] S. Gibson "Using linked volumes to model object collision, deformation cutting, carving, and joining," *IEEE Visualization and Computer Graphics*, pp.169-177, 1999.
- [24] E.Anshelevich, S.Owens, F.Lamiroux and L.Kavraki, "Deformable volumes in path planning applications," *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp.2290-2295, 2000.
- [25] F.Lamiroux and L.Kavraki, "Planning paths for elastic objects under manipulation constraints," *Int. J. Robotics Research*, Vol.20, No.3, pp.188-208, 2001.
- [26] O.B.Bayazit J.L.Nancy and M. Amato, "Probabilistic Roadmap Motion Planning for Deformable Objects," *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp.2126-2133, 2002.
- [27] M.C.Lin and J.F.Canny, "A Fast Algorithm for Incremental Distance Calculation," *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp.1008-1014, 1991.
- [28] J.-C.Latombe, "Robot motion planning", *Kluwer Academic Publishers.*, 1991.

Precise Deformation of Rheologic Object under MSD Models with Many Voxels and Calibrating Parameters

Ryo Nogami Hiroshi Noborio

Graduate School of Engineering, OECU
Hatsu-cho 18-8, Neyagawa, Osaka 572-8530, Japan

Email: nogami@noblabs.osakac.ac.jp nobori@noblabs.osakac.ac.jp

Fumiaki Ujibe Hiroki Fujii

Department of Engineering Informatics
Osaka Electro-Communication University

Email: ujibe@noblabs.osakac.ac.jp

Abstract—The MSD (Mass-Spring-Damper) model efficiently calculates shape deformation of many kinds of materials such as elastic, visco-elastic, and rheologic objects. For this reason, dynamic animation can be made in a personal computer and its popular acceleration board within the video-frame rate. The problem of MSD model is how to maintain shape precision of each deformation. For this purpose, we have calibrated coefficients of damper and spring of Voigt part and a coefficient of damper of the other part in the basic MSD element under many surface points capturing a real rheologic object by the randomized algorithm. Nevertheless, the shape precision is not unfortunately enough. To overcome this, we improve our previous approach in the following five points:

(1) The number of voxels in the MSD model increases from 75 to 600.

(2) The ratio between lengths of Voigt and the other parts in the MSD element is added to three coefficients of spring and dampers of the basic element as calibrating parameters.

(3) Four unknown parameters of the basic element are distinguished to calibrate in and on each voxel. In addition, the parameters are distinguished to calibrate among surface and core areas of a virtual rheologic object.

(4) In order to speed up the calibration, we use GA (Genetic Algorithm) in replace of RA (Randomized Algorithm).

(5) Each or both of local and global volume constant conditions are added into the previous approach.

In conclusion, we investigate relations between shape deformation, volume resolution, and number of calibrated parameters in several MSD models representing a rheologic object. Also, we improve deformation precision by increasing not only volume resolution but also number of calibration parameters or by adding each or both of volume constant conditions.

I. INTRODUCTION

Real-time dynamic animations of many kinds of deformable materials are developing because of wonderful PC, graphics acceleration board and so on. Modeling many types of deformable objects is indispensable in robotics and virtual reality. The key trade-off occurs between calculation time and deformation accuracy. The performance of modern computers and graphics hardware has made physically-based animation possible in real time. But even with today's best hardware and most sophisticated technique [1],[2],[3], only few hundred elements with small deformation have been simulated in real-time. Furthermore, the deformation is not always precise

because shape sequences of real and virtual objects are not compared yet.

In order to construct models of elastic and visco-elastic objects, we have usually used one of four methods, i.e., the mass-spring-damper (MSD) method [4],[5],[6],[7], the finite difference method (FDM) [8], the boundary element method (BEM) [9], and the finite element method (FEM) [10],[11],[12],[13],[14]. The computation cost and deformation accuracy increase in this order. To solve the trade-off, we have studied two approaches, so called fast FE and precise MSD methods. In this research, we focus on the latter method. Modeling a rheology object is more difficult than doing an elastic or visco-elastic object because the rheology object always leaves a residual displacement after pushing operation. For this reason, a rheology object has been seldom modeled except few works [15],[16],[17],[18],[19]. Also, whether the modeling is precise has not been estimated yet. Few papers calibrated uncertain parameters of elastic or visco-elastic virtual object by many experimental data captured from a real one [6],[7].

In this paper, we derive elastic, viscous, and rheologic properties by many deformations captured from a real rheology object. For this purpose, we extend our previous MSD model to three kinds of MSD models by adding local and/or global volume constant conditions. In order to maintain a better sequence of shape deformations, we increase a discrete number of voxels, and also distinguish to calibrate at most four set of uncertain coefficients in the MSD models under a lot of experimental data by GA (Genetic Algorithm) in replace of RA (Randomized Algorithm) as an optimal or near-optimal probabilistic search. Four sets are classified as follows: (1) One set is allocated for elements on a voxel, and another set is allocated for elements in the voxel. (2) One set is allocated for elements in core area of virtual rheology object, and another set is allocated for elements in surface area of it. Thirdly, we carefully compare four models in a popular personal computer with a general graphics acceleration board, and consequently select the best one for maintaining a better trade-off between shape precision and calculation time.

In this paper, section 2 describes our voxel/lattice structure

with many basic MSD elements which consist of Voigt and damper parts. In addition, we explain how to calculate shape deformation and force propagation in the structure. They are calculated by solving dynamic equation (quadratic differential equation). It is approximately calculated by many kinds of numerical methods for integrating ordinary differential equations. Moreover, by adding local and/or global volume constant conditions, we propose three extended models. Section 3 explains how to calibrate uncertain parameters in Voigt and damper parts and their length ratio. For this purpose, we use randomized and genetic algorithms as probabilistic near-optimal algorithm. In section 4, we prepare several kinds of comparative experimental results to select the best MSD model. Finally in section 5, we give a few conclusions.

II. THREE EXTENDED MASS-SPRING-DAMPER MODELS

In this section, we propose three extended mass-spring-damper (MSD) models. Firstly, we introduce our element with two dampers and one spring. Secondly, we explain four kinds of MSD models. The model 1 forms a basic voxel/lattice structure with three lengths of elements. In this model, we explain how to propagate internal forces. Finally, in order to decrease the volume difference between virtual and real rheology objects, we consider two types of modifications in the model 1. In the model 2, we replace four longest diagonal elements of eight half-length elements in each voxel of the model 1. By the replacement, each volume is strongly expanded. In the model 3, we add a set of external forces around the surface of virtual rheology object so as to maintain the volume of real rheology object. We call these as local and global volume constant conditions. Finally in the model 4, we construct the virtual rheology object including basic MSD local and global conditions.

A. Our Element with Voigt and Damper Parts

First of all, we introduce our basic MSD element which consists of Voigt and damper parts serially (Fig.1). This is similar to elements proposed in [15],[16]. The element in [15] consists of Voigt and Maxcell models serially, and that in [16] consists of Voigt model and adaptive damper serially. The adaptive damper flexibly controls coefficient of damper during pushing and releasing operations.

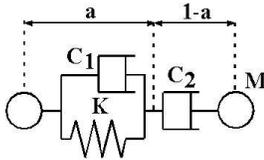


Fig. 1. Our basic MSD element with two dampers and one spring.

In our basic MSD element, the larger the coefficient K is, the stronger the elasticity is. Thus, K controls displacement of deformation behavior. The larger the coefficient C_1 is, the stronger the viscosity is. Therefore, C_1 controls speed of the behavior. Finally, the larger the coefficient C_2 is, the larger the residual displacement is. As long as C_2 is small enough, the object appears elastic or visco-elastic property. On the other

hand, if C_2 is large enough, the object appears plastic property. If K , C_1 or C_2 is too small, each element loses elastic, viscous, or rheologic property. Finally, the initial length of basic MSD element between neighbor mass points is denoted as 1, and the ratio between Voigt and damper parts is denoted as $a : 1 - a$ (Fig.1). If the length of Voigt part is too long, the residual replacement does not appear. If the length of damper part is too long, the back replacement does not appear. To avoid such cases, we set an interval $0.3 \leq a \leq 0.7$.

B. A Basic Voxel/Lattice Model

A rheology object deforms in a 3-D environment. In order to describe several kinds of deformations flexibly, we adopt a symmetric voxel/lattice structure to describe a rheology object [15],[16]. In the structure, let us distribute mass points uniformly in a rheology object whose intervals are the same along X, Y, and Z axes (Fig.2(a)). Let N be the number of mass points and M_{object} be the total mass of rheology object. Therefore, each mass point is given by $M = M_{object}/N$. In the experiment, we use $M_{object} = 864$ [g], $N = N_x \times N_y \times N_z$. For example, $M = 6.00$ for $N = 144 = 6 \times 4 \times 6$, or $M =$ about 1.02 for $N = 847 = 11 \times 7 \times 11$.

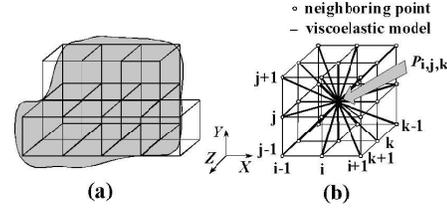


Fig. 2. (a) A voxel/lattice model of rheology object, (b) Neighboring mass points by basic MSD elements.

In our experiment, a real rheology object is built by mixing wheat flour and water. The rheology object whose scales are $10\text{cm} \times 6\text{cm} \times 10\text{cm}$ along X, Y and Z axes is horizontally and vertically two times larger than its virtual rheology object whose scales are $5\text{cm} \times 3\text{cm} \times 5\text{cm}$ along the axes. Therefore, the former volume (600 [cm^3] = water weight [g]) is eight times larger than the latter volume (75 [cm^3]). Based on this similarity, force propagation and shape deformation of real object equals to these of virtual one.

The elements are inserted between neighboring mass points as illustrated in Fig.2(b). The virtual rheology object is deformed by expanding and contracting the elements. Let $P_{i,j,k}$ be position vector corresponding to mass point (i, j, k) (i, j, k are integers, $N_x \geq i \geq 1$, $N_y \geq j \geq 1$ and $N_z \geq k \geq 1$, N_x, N_y, N_z : the numbers of masses along X, Y, Z axes, respectively). Let us derive quadratic differential equation of each mass at $P_{i,j,k}$. Each internal force acting on $P_{i,j,k}$ by the element between $P_{i,j,k}$ and its neighbor $P_{i+\alpha, j+\beta, k+\gamma}$ is denoted by $F_{i,j,k}^{\alpha,\beta,\gamma}$. For each mass, 6 shorter elements whose distance is denoted as l , 12 normal elements whose distance is denoted as $\sqrt{2}l$, and 8 longer elements whose distance is denoted as $\sqrt{3}l$ are located. As a result, the unit length l is defined as $5/(N_x - 1)$ cm in a virtual rheology object.

Therefore, total internal force $F_{i,j,k}^I$ acting on $P_{i,j,k}$ is given by the sum of 26(= 6 + 12 + 8) internal forces $F_{i,j,k}^{\alpha,\beta,\gamma}$. Moreover, if the sum of active external forces at $P_{i,j,k}$ is denoted by $F_{i,j,k}^E$, we obtain the following quadratic differential equation. This summation requires small calculation time.

$$M\ddot{P}_{i,j,k} = F_{i,j,k}^I + F_{i,j,k}^E \quad (1)$$

In order to calculate next position $P_{i,j,k}$ ($N_x \geq i \geq 1$, $N_y \geq j \geq 1$ and $N_z \geq k \geq 1$) at each mass, we should solve the above differential equation. This is done by many kinds of numerical integration methods, but they are too expensive.

$$F_{i,j,k}^I = \sum_{\substack{\alpha,\beta,\gamma \in \{-1,0,1\} \\ (\alpha,\beta,\gamma) \neq (0,0,0)}} F_{i,j,k}^{\alpha,\beta,\gamma} \quad (2)$$

We note that mass positions on an encountered surface between a rheology object and its pushed rigid body, and on the whole floor are fixed. Therefore, numerical integrations at the positions can be neglected.

C. Voxel/Lattice Model Including Feed Forward (Local) Volume Constant Condition

In our previous work, we understand shape of a calculated (virtual) rheology object is not the same against shape of its experimental (real) object. Especially, volume of the former object is too small against volume of the latter object after releasing. To overcome this, we expand volume of each voxel by local volume constant condition.

The local volume constant condition extends a voxel during deformation by eliminating four longest elements (whose distances are $\sqrt{3}$ cm in Fig.3(a)) and adding eight half-length elements from its center of gravity to eight vertices (whose distances are $\sqrt{3}l/2$ cm in Fig.3(b)).

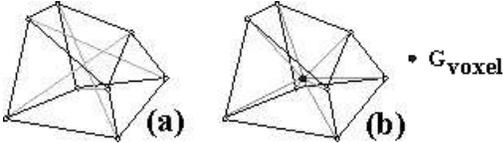


Fig. 3. Local volume constant condition: (a) Four longest elements directly connecting two opposite vertices in a voxel. (b) Eight shorter elements connecting from the center G_{voxel} of gravity of a voxel to its eight vertices.

This technique has been already used in a MS (mass-spring) model [20]. This controls isotropy or anisotropy of the elastic material. This idea can be straightforwardly extended to rheological material controlled by a MSD (mass-spring-damper) model.

D. Voxel/Lattice Model Including Feedback (Global) Volume Constant Condition

The local volume constant condition is a feed-forward approach and thus cannot decrease the volume difference between real and virtual objects directly. For this reason, the volume of virtual object does not equal to that of real one. Also, the local volume constant condition does not ensure stability of shape deformation, especially if all uncertain parameters are not well calibrated. To overcome this, we test

another volume constant condition which has been used in [16],[21],[22].

According to the global volume constant condition based on **Pascal's Principle**, we always add a set of external forces to a set of mass points around the surface of virtual rheology object by minimizing the volume difference between V_{int} and V (V_{int} : the volume of real rheology object, V : the volume of virtual rheology object). If $V \leq V_{int}$ is satisfied, an outward external force p appears around the virtual rheology object (Fig.4(a)), otherwise, an inward external force p appears around the object (Fig.4(b)) (\dot{V} : time difference of the volume V , K_{vol} : elastic coefficient of global volume constant condition, C_{vol} : viscous coefficient of global volume constant condition). The sum of magnitudes of external forces is denoted as p .

$$p = -K_{vol}(V - V_{int}) - C_{vol}\dot{V} \quad (3)$$

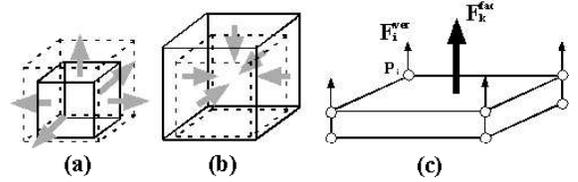


Fig. 4. Global volume constant condition: (a) If $V \leq V_{int}$ is satisfied, an outward external force appears around the virtual rheology object. (b) If $V_{int} \leq V$ is kept, an inward external force occurs around it. (c) An external force F_k^{fac} on a patch P_k is distributed into four forces F_i^{ver} of its corners.

Furthermore, concerning to a patch P_k whose number is k , area is S_k , and outward unit normal vector is n_k^{out} . By the global volume constant condition, there is an additional force F_k^{fac} at a patch P_k (Fig.4(c)).

The force F_k^{fac} acting to the patch P_k is denoted as the following equation.

$$F_k^{fac} = pS_k n_k^{out} \quad (4)$$

In succession, the force F_k^{fac} is distributed into four forces at corner masses of P_k . Therefore, the force F_i^{ver} acting to each corner mass is denoted by the following equation (i : mass number around a virtual rheology object, A_i : a set of all patches including the mass).

$$F_i^{ver} = \sum_{k \in A_i} \frac{F_k^{fac}}{4} \quad (5)$$

The sum of external forces at mass points around a virtual rheology object amounts to zero. This means that the set of internal forces is always constant and also momentum variation is completely fixed. Furthermore, as long as two coefficients K_{vol} and C_{vol} are set large enough, volumes of virtual and real rheology objects strictly equal each other. The parameter calibration is not necessary because of the feedback depending on their volume difference.

III. THE COMPARISON OF GA AND RA TO CALIBRATE MANY UNCERTAIN PARAMETERS

The defective point of *MSD* is accuracy of force propagation and shape deformation. To overcome this, we calibrate

many uncertain parameters of our MSD models by minimizing differences of shape deformations between real and virtual rheology objects. For the calibration, we use *RA* (*Randomized Algorithm*) and *GA* (*Genetic Algorithm*).

A. How to Calculate Shape Difference Between Real and Virtual Rheology Objects

In this research, a rheology object is precisely pushed by a rigid body located at the tip of a robotic manipulator (Fig.5(a)). The deformation, that is, the sequence of shapes is measured by two stereo vision camera systems *Digiclops* and its software development kit (SDK) *Triclops* (provided by Point Grey Research Inc, Canada). Each captures about three or more thousand points as shape of real rheology object in the real-time manner. After capturing the shape deformation, we finally measure how much total volume is changed before pushing and after releasing the rheology object. For the measurement, we use the following primitive method. First of all, we fill a ball with water, and then drop a deformed object into the ball (Fig.5(b)). Secondly, we gather overflowed water and measure its weight by a precise electric balance. As a result, we can understand volume of rheology object always decreases by about three percentages.

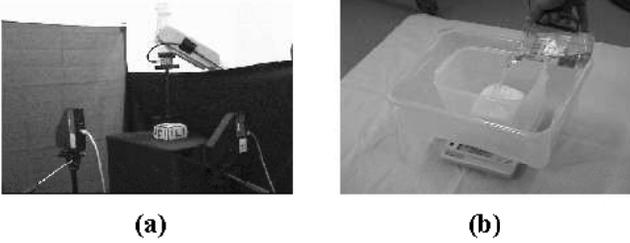


Fig. 5. (a)An experiment system: A real rheology object is pushed by a rectangular rigid body located at the tip of a robotic manipulator, and deformation of its sides is measured by two *Digiclops* cameras. (b)A photo when we measure volume of a rheology object before pushing and after releasing by overflowed water.

In order to evaluate a difference between real and virtual rheology objects, we summarize minimum distances from captured points to their nearest surfaces around a virtual rheology object. A virtual object consists of $N_x \times N_y \times N_z$ hexahedra which are individually deformed from initial cubes. Therefore in order to evaluate the difference, we calculate the minimum of $N_x \times N_y \times N_z$ shortest distances for each captured point to all hexahedra by the Lin-Canny closest point algorithm [23], and then we calculate sum S of the minimum distances for all captured points, which are larger than the average error $0.05cm$ of *Digiclops*. In our calibration, we use the sum of four S at four times during and after each pushing (Fig.6). By minimizing the total S in randomized or genetic algorithm, we can obtain a better set of all calibrated parameters.

B. Randomized Algorithm

1) *A Steepest Descendent Method*: **1.** Two parameters T_{cal} (threshold of calculation time) and T_{ran} (driving distance of random walk) are given in advance.

- 2.** Initialize all the uncertain coefficients within their individual intervals.
- 3.** We calculate shape difference S between real and virtual rheology objects.
- 4.** In order to find all the possible neighbors, we decrease and increase Δ at some of uncertain coefficients. For example, if the number of coefficients is 8 and 16, we have 2^8 and 2^{16} neighbors, respectively.
- 5.** If the minimum is smaller than S obtained in step 3, we move to the neighbor with the minimum (tie break arbitrary) by decreasing or increasing some coefficients by Δ , and then return to step 3. Otherwise, the algorithm finishes.

- 2) *Our Randomized Algorithm*: **1.** We randomly select a set of unknown coefficients and then calculate its value S as S_{ran} .
- 2.** By the steepest descendent method, we get one of the local minima, whose value is the smallest S . Then, we set S_{ran} by S . Then, if calculation time equals to or is large than T_{cal} , the algorithm ends, otherwise, move to step 3.
 - 3.** We randomly increase and decrease some of coefficients T_{ran} times by Δ , and then calculate its S . Then, if $S \leq S_{ran}$ is satisfied, we return to step 2, otherwise, continued to step 3.

C. GA (*Genetic Algorithm*)

- 1.** We give four parameters G_{ind} , G_{gen} , G_{eli} , G_{mut} . G_{ind} is the number of all individuals. Each consists of calibrated coefficients. G_{gen} is a given threshold of generation number. If the present generation P_{gen} amounts to this, GA finishes. G_{eli} is a survival ratio of all the individuals from the present generation to the next one ($0 \leq G_{eli} \leq 1$). Finally, G_{mut} is an arbitrary ratio of mutation.
- 2.** We construct all individuals whose number is G_{ind} , which are composed of all the calibrating coefficients. They are randomly numbered within their individual intervals. Moreover, we set the present generation P_{gen} as 0.
- 3.** We calculate shape differences $S_n (n = 1, 2, \dots, G_{ind})$ between real and virtual objects for all individuals.
- 4.** If $P_{gen} = G_{gen}$ is satisfied, the algorithm finishes.
- 5.** After sorting all individuals in the order of S_n , we select higher individuals whose number is $G_{ind} \times G_{eli}$. Then we eliminate the others. We call this as selection.
- 6.** We randomly select two individuals A and B from all ones for crossing. First of all, each of two is bit-sized, and its crossing point is randomly selected. Secondly, we cut each at the crossing point to generate former parts A_f and A_l , and do latter parts B_f and B_l . Then, one former and another latter are combined as $A_f + B_l$. This operation is called as crossing. Moreover, we generate a new individual whose bits are reversed from an original individual. This rarely occurs by the probability G_{mut} . This operation is called as mutation.

Finally, $G_{ind} \times (1 - G_{eli})$ individuals newly appear in the next generation by crossing and mutation ($G_{ind} \times G_{eli}$ individuals are left from the last generation). Then, we increment P_{gen} .

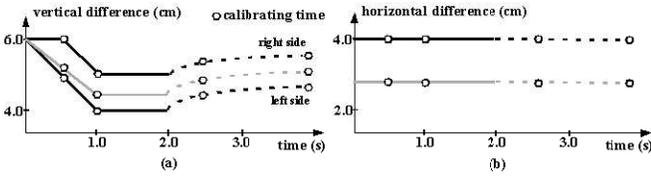


Fig. 6. The vertical and horizontal differences during two pushing operations are synchronously described as gray and black whole or dot line (Black line: First pushing, Gray line: Second pushing). (a) Y-axis direction. (b) X-axis direction.

IV. SEVERAL COMPARATIVE RESULTS

In this section, (1) concerning to shape accuracy and computation efficiency, we select the best numerical integration method from many kinds of methods such as *Runge – Kutta – Gill*, *Runge – Kutta*, *Midpoint*, *BDF*, *Euler* and *Implicit – Euler*. (2) concerning to computation time, memory storage and shape accuracy, we compare four MSD models with each other. The model 1 is a basic voxel/lattice MSD model. In the models 2, 3, 4, we add explained local, global, local and global volume constant conditions in the model 1. The deformation of virtual rheology object is calculated and visualized by a 3-D graphics software OpenGL in a personal computer (CPU: Pentium4 3.00GHZ, Main memory: 2048MB) with a 3-D graphics acceleration board (GeForce FX 5600, 128MB).

In all models, we distinguish MSD elements in surface and core areas of virtual rheology object to calibrate (Fig.7). The first set is K^{surf} , C_1^{surf} , C_2^{surf} , a^{surf} within the surface area, and the second set is K^{core} , C_1^{core} , C_2^{core} , a^{core} within the core area. In addition, in the models 2 and 4, we distinguish MSD elements on and in each voxel. That is, we categorize shortest 8 diagonal elements illustrated in Fig.3 as elements in a voxel, and categorize the other longer 18 elements as elements on a voxel. Therefore, the first set is $K^{surf-in}$, $C_1^{surf-in}$, $C_2^{surf-in}$, $a^{surf-in}$ in each voxel within surface area, the second set is $K^{core-in}$, $C_1^{core-in}$, $C_2^{core-in}$, $a^{core-in}$ in each voxel within core area, the third set is $K^{surf-on}$, $C_1^{surf-on}$, $C_2^{surf-on}$, $a^{surf-on}$ on each voxel within surface area, and the fourth set is $K^{core-on}$, $C_1^{core-on}$, $C_2^{core-on}$, $a^{core-on}$ on each voxel within core area. Consequently, we construct models 2' and 4' extended from models 2 and 4. Finally, Furthermore, in order to form a virtual object pushed by a rigid body stably (Fig.8(a)), we give the following intervals $50 \leq K^{surf-in}, K^{core-in} \leq 3000$, $100 \leq K^{surf-on}, K^{core-on} \leq 3000$, $250 \leq C_1^{surf-in}, C_1^{core-in} \leq 10000$, $500 \leq C_1^{surf-on}, C_1^{core-on} \leq 10000$, $250 \leq C_2^{surf-in}, C_2^{core-in} \leq 20000$, $500 \leq C_2^{surf-on}, C_2^{core-on} \leq 20000$, $0.3 \leq a^{surf-in}, a^{core-in}, a^{surf-on}, a^{core-on} \leq 0.7$. If some of them are too large, each element becomes unstable and consequently the shape of rheology model is crushed as Fig.8(b).

A. Relationship between Shape Precision and Integration Method

As mentioned in the section 2, in order to get position of mass point, we need to solve dynamic equations (1) and (2) including internal and external forces at the mass point. In this paragraph, similar to cloth animations [24], [25], we compare qualities of shape deformations by selecting many integration

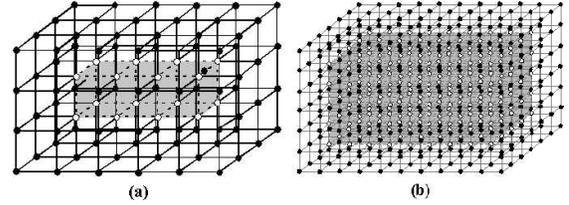


Fig. 7. Surface and core areas of virtual rheology object. Black and white mass points are located on the surface and core areas, respectively. Gray voxel and dotted line are among the core area. (a) $5 \times 3 \times 3$ voxels, (b) $10 \times 6 \times 6$ voxels.

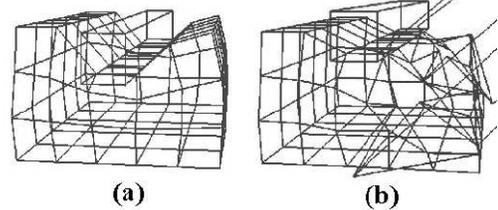


Fig. 8. (a) A stable shape of rheology object. (b) An unstable shape of the object.

methods such as *Runge–Kutta–Gill*, *Runge–Kutta*, *Midpoint*, *BDF*, *Euler* and *Implicit – Euler* in order to solve the equations (1) and (2). In order to keep the comparison fairness, we fix $K = 500$, $C_1 = 2000$, $C_2 = 2000$, $a = 0.5$ in the model 1 pushed by the first operation (Fig.9(a)). In addition, even though we evaluate the difference between virtual and real objects by S^2 , S^3 , S^4 (S : distance sum of captured points between virtual and real objects), all shape deformations have similar qualities. Here, since almost all the distances between captured points in a real object and their ones in a virtual object are within an interval $[0.0, 0.5]$ cm, we always keep the inequalities $S > S^2 > S^3 > S^4$. As shown in Table I and Fig.10, we can see shape differences of virtual and real rheology objects are similar in all the numerical integration methods. We suppose the reason is that the sampling time solving the equation is too small. As contrasted with this, as illustrated in Table I, if the numerical integration methods to solve the dynamics of each mass point are changed, their calculation times are drastically altered. In results, by replacing the *Runge – Kutta* method of the *Euler* method, we can save calculation time to solve the differential equation about four times. For this reason, we select the simpler method such as *Euler*, which exactly maintains shape precision by few calculation time.

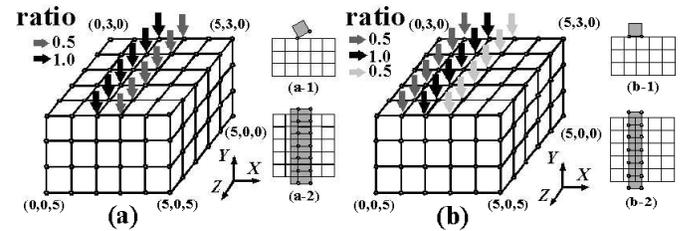


Fig. 9. (a) 3-D view for the first pushing. (a-1) Front view in (a). (a-2) Upper view in (a). (b) 3-D view for the second pushing. (b-1) Front view in (b). (b-2) Upper view in (b).

TABLE I

SHAPE PRECISION AND CALCULATION TIME BY INTEGRAL METHODS FOR THE FIRST PUSHING IN A BASIC Voxel/LATTICE MODEL.

Compare indexes	S [cm]	S^2 [cm ²]	S^3 [cm ³]	S^4 [cm ⁴]	T_{int} [sec]
Runge-Kutta-Gill	2508.076	608.875	168.545	51.292	17.664
Runge-Kutta	2508.076	608.875	168.545	51.292	17.297
Midpoint	2565.494	633.363	177.656	54.221	6.939
BDF	2519.145	613.609	170.393	52.008	6.977
Euler	2518.455	613.325	170.280	51.963	6.891
Implicit-Euler	2496.485	603.153	165.668	49.534	10.297

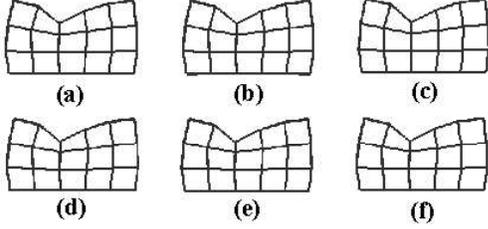


Fig. 10. The virtual rheology object deformed by many integral methods. (a) Runge-Kutta-Gill (b) Runge-Kutta (c) Midpoint (d) BDF (e) Euler (f) Implicit-Euler.

B. Computation Complexity and Memory Storage

In the virtual 3-D graphics world, we construct a basic voxel/lattice model called as the model 1, and build three extended models with/without local and global volume constant conditions as the models 2, 3 and 4. First of all, we generate 2000 deformations of virtual rheology object in simulation, which correspond to their deformations of real rheology object in all experimental trials by 4 [s]. Therefore, each deformation time in simulation corresponds to 2 [ms] in experiment. Secondly in each simulation, we make their deformations of virtual rheology object by about 8 [s] in the models 1 and 2 or 14 [s] in the models 3 and 4 whose size is $6 \times 4 \times 6$. Therefore, PC and graphics acceleration board require about 4 or 7 [ms] per each deformation (Table II). Since they are smaller than the video-frame rate (33[msec]), we can watch dynamic animation in real-time.

In the models 1 and 2, total time T_{def} for calculating deformation almost equals to partial time T_{int} for integrating many mass points, i.e., $T_{def} = T_{int}$. T_{int} in four models directly depends on number of mass points $N_{cal} = (N_x \times N_y \times N_z) - f_{out} - (N_x \times N_z)$ (N_x, N_y, N_z : numbers of masses along X, Y, Z axes, respectively, f_{out} : number of masses pushed by the rigid object). As shown in Table II, since $f_{out} = 12$ is for the first pushing under $6 \times 4 \times 6$ model, number of masses to need the integration is denoted as $N_{cal} = (6 \times 4 \times 6) - 12 - (6 \times 6) = 96$. Similarly, mass numbers to need the integration are denoted as 704, 2272 and 5250 for $11 \times 7 \times 11$, $16 \times 10 \times 16$ and $21 \times 13 \times 21$ resolutions, respectively. As described in Table II, calculation time of each model is proportionally evaluated by the equation $N_{cal} = (N_x \times N_y \times N_z) - f_{out} - (N_x \times N_z)$. Furthermore, in the models 3 and 4, we should additionally consider time T_{vol} to calculate volume of virtual rheology object by summing up volumes of its polyhedrons. T_{vol} directly depends on $L_{cal} = (N_x - 1) \times (N_y - 1) \times (N_z - 1)$. For this reason, in the models 3 and 4, time T_{def} to calculate each deformation equals to total time T_{int} and T_{vol} , i.e., $T_{def} = T_{int} + T_{vol}$.

Since numbers of mass points in four kinds of models

are almost the same, memory storage is also the same. The memory storage m is approximately evaluated by summing m_S and m_N . The mass storage m_N directly depends on the number of masses, and the software storage m_S is determined as the sum of C++ compiler (VC++ 6.0) and OpenGL software. It is invariable. Using results in Table II, we calculate $m_S = 33[MB]$ (C++ compiler = 20[MB] and OpenGL = 13[MB]) and $m_N = 12 \times N[KB]$.

TABLE II

CALCULATION TIME [msec] AND MEMORY STORAGE [MB] PER ONE DEFORMATION IN PC.

Total calculation time for the first pushing [msec]				
	[6 × 4 × 6]	[11 × 7 × 11]	[16 × 10 × 16]	[21 × 13 × 21]
(model 1)	3.617	30.211	74.578	169.282
(model 2)	4.418	31.829	79.445	174.352
(model 3)	7.235	53.156	153.672	337.789
(model 4)	7.570	54.328	158.898	350.641
Deformation calculation time for the first pushing [msec]				
(model 1)	2.063	15.023	47.953	117.342
(model 2)	2.494	16.470	54.608	127.637
(model 3)	5.501	37.968	129.651	290.467
(model 4)	5.820	39.141	134.482	303.230
Memory storage for the first pushing [MB]				
(model 1)	34.744	43.492	62.700	99.344
(model 2)	34.240	43.512	62.688	99.112
(model 3)	34.044	43.644	62.680	99.372
(model 4)	34.264	44.884	62.904	99.168

C. Comparison between RA and GA

In this paragraph, we compare efficiencies and qualities of RA (Randomized Algorithm) and GA (Genetic Algorithm) to calibrate a better set of uncertain parameters in the model 1 which consists of $5 \times 3 \times 5$ voxels. Firstly, we set parameters $\Delta = 10$, $T_{cal} = 168[hour]$ (1week), and $T_{ran} = 100[number]$ in RA. In order to calculate the sum of differences between real and virtual rheology objects at all the captured points, we need about 8 seconds and consequently investigate sets of uncertain parameters, whose numbers are about 75600. Secondly, we compare RA with GA fairly. For this purpose, we calibrate another better set of uncertain parameters by GA. Firstly, we set G_{ind} as a small value 50. The reason is similar individuals appear if G_{ind} is larger. Then, we set G_{eli} as a small value 0.2 because of the same reason. Thirdly, we set G_{gen} as 1900. The reason is . In RA, we check many points whose number is about 75600. After determining $G_{ind} = 50$ and $G_{eli} = 0.2$, we always get 40 individuals in each generation. Therefore, we should pass though 1900 generations so as to get 75600 individuals. Finally, since G_{mut} should be selected as a smaller value, we set G_{mut} as 0.01.

The calibrations of eight uncertain parameters in RA and GA are compared in Table III. The pushing operation is the same (called as the first pushing). All calibrated parameters, error distance and volume precision between real and virtual rheology objects are described in Table III. Because of their shape and volume consistencies, we understand GA is better than RA for the calibration. The reason is as follows: RA is a probabilistic search which randomly and sequentially selects initial points to investigate their local minima. Then,

it finally selects the smallest value of many local minima as a near-optimal solution. Therefore, RA consists of many local searches. As contrasted with this, GA synchronously selects and extends many better points in each generation. Therefore, GA simultaneously switches global and local investigations. As a result, GA quickly finds a hopeful area even though search space is too huge.

TABLE III

CALIBRATION RESULTS BY RANDOMIZED AND GENETIC ALGORITHMS FOR THE FIRST PUSHING IN MODEL1. (S : THE SUM OF ERROR DISTANCES)

The number of captured points is $N = 14551$					
Calibration result	S [cm]	Volume [cm^3]		S [cm]	Volume [cm^3]
(model 1-RA)	2442	67.35	(model 1-GA)	2350	67.85
		K [gf/cm^3]	C_1 [gfs/cm^3]	C_2 [gfs/cm^3]	a
(model 1-RA) - core		2245	4970	4277	0.68
(model 1-RA) - surf		1870	3026	2526	0.47
(model 1-GA) - core		293	9988	19961	0.63
(model 1-GA) - surf		2477	561	4454	0.63
The number of points whose errors are larger than 0.25cm					
	first	second	third	fourth	total
(model 1-RA)	1182	922	721	737	3562
(model 1-GA)	1091	990	579	637	3297

D. Comparison between Sparse and Dense Four Models

In this paragraph, in order to check shape precision in the progression of model resolution, we prepare two MSD models whose sizes are $6 \times 4 \times 6$ and $11 \times 7 \times 11$. All calibrated parameters, error distance and volume precision between real and virtual rheology objects are described for models 1 ~ 4 whose sizes are $6 \times 4 \times 6$ and $11 \times 7 \times 11$ in Table IV and Table V, respectively. In addition, shape deformations after pushing and releasing are illustrated in Fig.11 and Fig.12 for models 1 ~ 4' whose sizes are $6 \times 4 \times 6$ and $11 \times 7 \times 11$, respectively. Although all are for the second pushing operation (Fig.9(b)), tendency is the same for the first pushing operation (Fig.9(a)).

TABLE IV

CALIBRATION RESULTS BY GENETIC ALGORITHM FOR THE SECOND PUSHING IN THREE MODELS (S : THE SUM OF ERROR DISTANCES).

The number of captured points is $N = 15372$					
Calibration result	(model 2)	(model 3)	(model 4)	(model 2')	(model 4')
S [cm]	2261	1724	1855	1831	1835
Volume [cm^3]	69.62	74.97	74.49	76.30	75.94
K^{core} [gf/cm^3]	2986	2994	104	220	137
C_1^{core} [gfs/cm^3]	508	1262	9839	1030	620
C_2^{core} [gfs/cm^3]	19149	12839	4071	522	1747
a^{core}	0.66	0.70	0.70	0.60	0.60
K^{surf} [gf/cm^3]	2232	3000	1687	313	142
C_1^{surf} [gfs/cm^3]	1980	667	559	1008	630
C_2^{surf} [gfs/cm^3]	13121	6457	9169	505	1938
a^{surf}	0.66	0.70	0.30	0.60	0.61
		K^{-in} [gf/cm^3]	C_1^{-in} [gfs/cm^3]	C_2^{-in} [gfs/cm^3]	a^{-in}
(model 2') -core		2901	1163	13672	0.45
(model 2') -surf		2880	1302	14888	0.45
(model 4') -core		2286	920	16399	0.64
(model 4') -surf		2301	910	16410	0.70
The number of points whose errors are larger than 0.25cm					
	first	second	third	fourth	total
(model 2)	486	545	910	961	2902
(model 3)	361	265	299	267	1192
(model 4)	314	563	494	426	1797
(model 2')	187	395	415	393	1390
(model 4')	215	331	480	567	1593

TABLE V

CALIBRATION RESULTS BY GENETIC ALGORITHM FOR THE SECOND PUSHING IN THREE MODELS (S : THE SUM OF ERROR DISTANCES).

The number of captured points is $N = 15372$					
Calibration result	(model 2)	(model 3)	(model 4)	(model 2')	(model 4')
S [cm]	1518	1663	1383	1506	1352
Volume [cm^3]	71.14	75.16	74.58	72.88	75.64
K^{core} [gf/cm^3]	2990	2796	1903	2424	287
C_1^{core} [gfs/cm^3]	3194	536	2688	4901	1961
C_2^{core} [gfs/cm^3]	4118	525	807	2664	5559
a^{core}	0.69	0.40	0.30	0.61	0.59
K^{surf} [gf/cm^3]	1781	2704	2232	1899	1313
C_1^{surf} [gfs/cm^3]	3603	539	732	5837	3895
C_2^{surf} [gfs/cm^3]	1735	9177	3401	10606	8416
a^{surf}	0.69	0.47	0.61	0.47	0.70
		K^{-in} [gf/cm^3]	C_1^{-in} [gfs/cm^3]	C_2^{-in} [gfs/cm^3]	a^{-in}
(model 2') -core		1902	2015	1759	0.31
(model 2') -surf		236	603	19003	0.59
(model 4') -core		117	632	1875	0.63
(model 4') -surf		412	1300	2783	0.53
The number of points whose errors are larger than 0.25cm					
	first	second	third	fourth	total
(model 2)	150	111	252	291	804
(model 3)	523	263	199	212	1197
(model 4)	157	261	425	165	1008
(model 2')	219	167	232	267	885
(model 4')	258	188	168	147	761

Times differ each other in the first pushing, which left and right edges of a rigid body contact a rheology object. Therefore, a set of mass points around the left edge is firstly affected by a set of external forces, and secondly another set of mass points around the right edge is affected by another set of external forces. Because of digitalization of our models and time lag, all external forces are inaccurate. Moreover, each of edges is far from any mass point in the second pushing. For this reason, larger forces are affected at all mass points along $X=2$ and $Y=3$, which rigid and rheology objects encounter, relatively smaller forces are affected at all mass points along $X=1$ and $Y=3$ or $X=3$ and $Y=3$, which rigid and rheology objects do not encounter in $6 \times 4 \times 6$ resolution (Fig.9(b)). Because of discreteness of our models, it is difficult to distribute all the external forces precisely. This imprecision can be partially solved by increasing resolution of each model. For this reason, all models whose voxels are $11 \times 7 \times 11$ are more precise than these whose voxels are $6 \times 4 \times 6$.

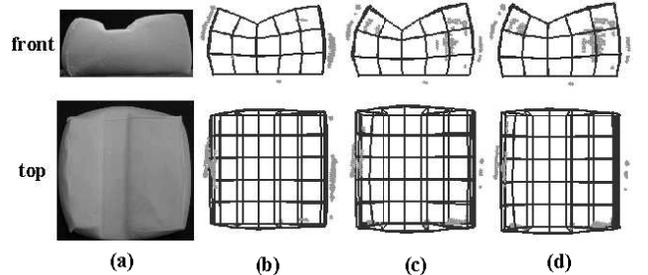


Fig. 11. The sum of differences between real and virtual rheology objects for the second pushing. (a) Real rheology object. (b),(c),(d) Virtual rheology objects. Areas are colored by dark gray, whose errors are larger than 0.25 [cm] in the models 3, 2' and 4'.

V. CONCLUSIONS

In this paper, in order to acquire more precise deformation of our voxel/lattice structure which consists of mass-spring-

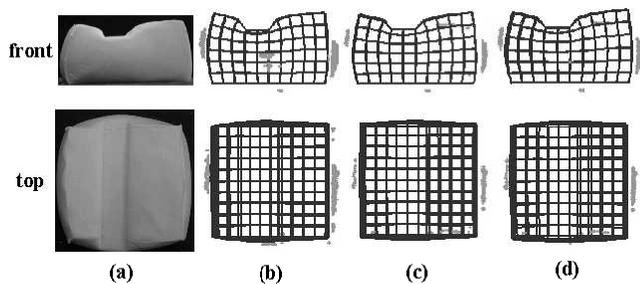


Fig. 12. The sum of differences between real and virtual rheology objects for the second pushing. (a) Real rheology object. (b),(c),(d) Virtual rheology objects. Areas are colored by dark gray, whose errors are larger than 0.25 [cm] in the models 3, 2' and 4'.

damper elements, we tried to test three ideas. First idea is to distinguish and calibrate many types of our basic MSD elements. Second idea is to increase volume resolution, that is, number of voxels in a virtual rheologic object. Third idea is to introduce two kinds of volume constant conditions, that is, local (feed-forward) and global (feedback) ones. The concluding remarks are as follows:

(1) The larger the number of calibrating parameters is, the larger the calculation time to calibrate is, but the more precise shape of the virtual rheology object is.

(2) The larger the number of voxels is, the larger the calculation time to form is, but the more precise the shape of virtual rheology object is.

(3) The local volume constant condition leads a feed-forward action. This is quick but unstable. Therefore, if we spend much time to calibrate, a virtual rheology object quickly converges to adequate shape, otherwise, its shape is destroyed. On the other hand, the global volume constant condition leads a feedback action. This is slow but stable. Even though we cannot have much time to calibrate, the virtual rheology object frequently converges to reasonable shape. This defective point is time consuming because of calculating the volume of virtual rheology object.

As a result, if the model resolution is lower, we need the global volume constant condition to converge shape of a virtual object. Therefore, we select the model 3 as the best one in $6 \times 4 \times 6$ resolution. Although tough calibration is necessary, the calculation is not so large. The reason is because deforming the virtual object is within a few milli-seconds. As contrasted with this, if the model resolution is higher, we require much computation costs to calibrate and deform. To save them, we use only the local volume constant condition which is about two times faster than the global one. For this reason, we select the model 2 or 2' as the best one in $11 \times 7 \times 11$ resolution. In this case, we spend much time to calibrate uncertain parameters carefully. Also, we should deform a virtual object by about thirty milli-seconds within the video-frame rate.

ACKNOWLEDGMENT

The authors thank Professor S.Hirai and H.Tanaka (Ritsumeikan Univ.) for discussions. This is supported in part by 2003 Grants-in-aid for Scientific Research from the Ministry of Education, Science and Culture, Japan (No.14550247).

REFERENCES

- [1] Y.Zhuang, *Real-time simulation of physically-realistic global deformations*. UC Berkeley: Doctoral Thesis, 2000.
- [2] G.Debunne, M.Desbrun, M. Cani, and A.Barr, "Dynamic real-time deformations using space and time adaptive sampling," in *Proc. SIGGRAPH Computer Graphics*, 2001, pp. 31–36.
- [3] X.Wu, M.S.Downes, T.Goktekin, and F.Tendick, "Adaptive nonlinear finite elements for deformable body simulation using dynamic progressive meshes," in *Proc. Eurographics Computer Graphics Forum*, 2001, pp. 349–358.
- [4] K.Waters, "A muscle model for animating three-dimensional facial expression," in *Proc. SIGGRAPH Computer Graphics*, 1987, pp. 17–24.
- [5] E.Promayon, P.Bacconnier, and C.Puech, "Physically-based deformations constrained in displacements and volume," in *Proc. Eurographics Computer Graphics Forum*, 1996, pp. 155–164.
- [6] J.Louchet, X.Provot, and D.Crochemore, "Evolutionary identification of cloth animation models," in *Proc. Eurographics Computer Animation and Simulation*, 1995, pp. 44–54.
- [7] A.Joukhadar, F.Garat, and C.Laugier, "Parameter identification for dynamic simulation," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 1997, pp. 1928–1933.
- [8] D.Terzopoulos, J.Platt, A.Barr, and K.Fleischer, "Elastically deformable models," in *Proc. SIGGRAPH Computer Graphics*, 1987, pp. 205–214.
- [9] D.James and D.K.Pai, "Artdefo, accurate real time deformable objects," in *Proc. SIGGRAPH Computer Graphics*, 1999, pp. 65–72.
- [10] M.B.Nielsen and S.Cotin, "Real-time volumetric deformable models for surgery simulation using finite elements and condensation," in *Proc. Eurographics Computer Graphics Forum*, 1996, pp. 57–66.
- [11] M.Muller, J.Dorsey, L.McMillan, R.Jagnow, and B.Cutler, "Stable real-time deformations," in *Proc. ACM SIGGRAPH Symposium on Computer Animation (SCA)*, 2002, pp. 49–54.
- [12] J.J.Berkley, *Haptic rendering of deformable bodies using real-time finite element analysis: an application to surgical simulation*. University of Washington: Doctoral Thesis, 2002.
- [13] S.P.DiMaio, *Modelling, simulation and planning of needle motion in soft tissues*. University of British Columbia: Doctoral Thesis, 2003.
- [14] H.-W. Nienhuys, *Cutting in deformable objects*. Utrecht University: Doctoral Thesis, 2003.
- [15] S.Tokumoto, Y.Fujita, and S.Hirai, "Deformation modeling of viscoelastic objects for their shape control," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 1999, pp. 1050–1057.
- [16] S.Tokumoto, S.Hirai, and H.Tanaka, "Constructing virtual rheological objects," in *Proc. World Multiconference on Systemics Cybernetics and Informatics*, Auland, July 2001, pp. 106–111.
- [17] M.Kimura, Y.Sugiyama, S.Tomokuni, and S.Hirai, "Constructing rheologically deformable virtual objects," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2003, pp. 3737–3743.
- [18] H.Noborio, R.Enoki, S.Nishimoto, and T.Tanemura, "On the calibration of deformation model of rheology object by a modified randomized algorithm," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2003, pp. 3729–3736.
- [19] H.Noborio, R.Nogami, and R.Enoki, "Precise deformation of rheology msd model calibrated by randomized algorithm," in *Eurographics2003 (Short Presentations)*, 2003, pp. 171–178.
- [20] D.Bourguignon and M.-P.Cani, "Controlling anisotropy in mass-spring systems," in *Proc. of the 11th Eurographics Workshop on Computer Animation and Simulation*, 2000, pp. 113–123.
- [21] I.Costa and R.Balaniuk, "Static solution for realtime deformable objects with fluid inside," in *In ERCIM News*, 2001, pp. 44–45.
- [22] K.Sundaraj, C.Laugier, and I.F.Costa, "An approach to lem modelling : construction, collision detection and dynamic simulation," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2001, pp. 2196–2201.
- [23] M.C.Lin and J.F.Canny, "A fast algorithm for incremental distance calculation," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 1991, pp. 1008–1014.
- [24] D.Baraff and A.Witkin, "Large steps in cloth simulation," in *Proc. SIGGRAPH Computer Graphics*, 1998, pp. 43–54.
- [25] M.Hauth, O.Etzmuss, B.Eberhardt, R.Klein, R.Sarlette, M.Sattler, K.Daubert, and J.Kautz. (2002) Cloth animation and rendering. Eurographics 2002 (Tutorial). [Online]. Available: <http://www.eg.org/EG/DL/Conf/EG2002/tutorials>

A Comparative Study of Rheology MSD Models whose Structures are Lattice and Truss

Ryo Nogami Hiroshi Noborio
Graduate School of Engineering
Osaka Electro-Communication University
Neyagawa, Osaka 572-8530, Japan
e-mail: nobori@noblab.osakac.ac.jp

Seiji Tomokuni Shinichi Hirai
Graduate School of Engineering Science
Ritsumeikan University
Kusatsu, Shiga 525-8577, Japan
e-mail: hirai@se.ritsumei.ac.jp

Abstract—In this paper, we compare two major structures of MSD (Mass-Spring-Damper) particle models. One is the lattice (hexahedral) structure, and the other is the truss (tetrahedral) structure. They (especially, the truss structure) have been frequently used for representing elastic and/or visco-elastic object. The MSD model efficiently calculates shape deformation of the above materials. In addition, in order to maintain shape precision of each deformation, we carefully calibrate coefficients of damper and spring of Voigt part and a coefficient of damper of the other part in the basic MSD element under many surface points capturing a real rheology object. A genetic algorithm is used for probabilistic calibration. After the comparison, we get the following properties:

(1) The lattice structure has too many elements for calculating force propagation. Therefore, it precisely leads shape deformation with the help of the local (feed-forward) volume constant condition.

(2) The truss structure does not have enough elements for propagating internal forces. Therefore, in order to keep a reasonable volume by expanding its virtual rheology object, we need the global (feed-back) volume constant condition.

(3) The global condition is time consuming, but can directly control the total volume of virtual rheology object. On the other hand, the local one is quick, but directly expands only a part (voxel) of the virtual object. Therefore, the volume and shape in the lattice structure with the local condition are better than those in the truss structure including the global one.

(4) The number of MSD elements in the lattice structure is about two times larger than that in the truss one. Therefore, the former calculation is about two times slower than the latter one. As contrasted with this, the global volume constant condition is strictly two times or more slower than the local one. As a result, calculation time of the lattice structure with the local condition is smaller than that of the truss structure with the global one.

In conclusion, the lattice structure with the local volume constant condition is the best concerning to calculation cost and shape precision.

I. INTRODUCTION

Real-time dynamic animations of many kinds of deformable materials are developing because of wonderful PC, graphics acceleration board and so on. Modeling many types of deformable objects is indispensable in many areas such as tele-operation, humanoid, assembly and task planning, computer animation, game/amusement and so on. The key trade-off occurs between calculation time and deformation accuracy. The performance of modern computer

and graphics hardware has made physical-based animation in real-time. But even with today's best hardware and most sophisticated technique [1],[2],[3], only few hundred elements with small deformations have been simulated in real-time. Furthermore, the deformation is not always precise because shape sequences of real and virtual objects are not compared yet.

In order to model elastic or visco-elastic material, we have usually selected one of four methods, i.e., the mass-spring-damper (MSD) method [4],[5],[6],[7], the finite difference method (FDM) [8], the boundary element method (BEM) [9], and the finite element method (FEM) [10],[11],[12],[13],[14]. The computation cost and deformation accuracy increase in this order. To solve such a trade-off between computation cost and shape accuracy, we have studied two approaches, so called fast FEM and precise MSD approaches.

In this research, we focus on the latter approach to deal with a rheology object. Modeling a rheology object is more difficult than doing an elastic or visco-elastic object because the rheology object always leaves a residual displacement after pushing operation. For this reason, a rheology object has been seldom modeled except few works [15],[16],[17],[18],[19]. Therefore, whether the modeling is precise has not been estimated yet. Also, few papers dealt with calibration of a virtual elastic or visco-elastic object by many experimental data captured from its real one [6],[7].

In this paper, we derive elasticity and/or viscosity calibrated from deformations capturing a real rheology object [18],[19]. In this situation, we compare two MSD models whose structure is lattice with two MSD models whose structure is truss. So far, the truss structure [2],[10],[17],[20] has been frequently adopted against the lattice structure [15],[16],[18],[19]. In general, the number of MSD elements in the lattice structure is about two times larger than that in the truss structure. For this reason, force propagation and shape deformation are usually better in the lattice structure. To overcome this drawback in the truss structure, we should add the global (feed-back) volume constant condition. Furthermore, we compare MSD lattice structures with/without the local (feed-forward) volume constant condition and MSD truss structures with/without the global (feed-back) volume constant condition.

In this paper, section 2 firstly describes our lattice structure with many basic MSD elements which consist of Voigt and damper parts. In addition, we explain how to calculate shape deformation and force propagation in the structure. They are calculated by solving dynamic equation (quadratic differential equation). In this research, we select the classic Euler method as a numerical integration method for the ordinary differential equation. Moreover, we explain our truss structure with the MSD elements, and by adding local and global volume constant conditions into the lattice and truss structures, we mention new lattice and truss structures, respectively. Section 3 illustrates how to calibrate uncertain parameters in Voigt and damper parts and their length ratio. For this calibration, we use a genetic algorithm as a probabilistic near-optimal search. In section 4, we give several kinds of comparative experimental results to investigate the best MSD model and its properties. Finally in section 5, we give a few conclusions and ongoing remarks.

II. FOUR STRUCTURES CONSIST OF MSD ELEMENTS

In this section, we explain four kinds of structures including mass-spring-damper (MSD) elements. Each element consists of two dampers and one spring. The lattice structure without any volume constant condition is a basic structure with three lengths of the basic elements. In this model, we explain how to propagate internal forces in each rheology object. Moreover, we illustrate the truss structure without any volume constant condition by eliminating the longest elements from the lattice structure without any volume constant condition. Furthermore, in order to decrease the shape and/or volume difference between virtual and real rheology objects, we modify the lattice and truss structures without any volume constant condition into the lattice structure with the local one and the truss structure with the global one. Here in the local condition, we add a set of internal force to expand or reduce each voxel. Also in the global condition, we add a set of external forces whose sum equals to zero, which are around virtual rheology object for maintaining the volume of real rheology object.

A. Our Element with Voigt Model and Damper

First of all, we introduce our basic MSD element which consists of Voigt and damper parts serially (Fig.1). This is similar to elements proposed in [15],[16]. The element in [15] consists of Voigt and Maxcell models serially, and that in [16] consists of Voigt model and adaptive damper serially. The adaptive damper flexibly controls coefficient of the damper during pushing and releasing operations.

In our basic MSD element, the larger the coefficient K is, the stronger the elasticity is. Thus, K controls the displacement of deformation behavior. The larger the coefficient C_1 is, the stronger the viscosity is. Therefore, C_1 controls the speed of behavior. Finally, the larger the coefficient C_2 is, the larger the residual displacement is. As long as C_2 is small enough, the object appears elastic or visco-elastic property. On the other hand, if C_2 is large enough, the object appears plastic property. If K , C_1 or

C_2 is too small, each element loses elastic, viscous, or rheologic property. Finally, the initial length of basic MSD element between neighbor mass points is denoted as 1, and the length ratio between Voigt and damper parts is denoted as $a : 1 - a$ (Fig.1). If the length of Voigt part is too long, the residual replacement does not appear. If the length of damper part is too long, the back behavior does not appear. To avoid such cases, we set an interval $0.3 \leq a \leq 0.7$.

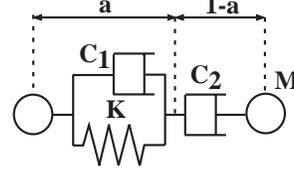


Fig. 1. Our basic element with two dampers and one spring.

B. A Basic Lattice (Hexahedral) Structure

A rheology object deforms in a 3-D environment. In order to describe several kinds of shape deformations flexibly, we firstly divide the real object into a set of voxels, and then allocate a lot of 1-D MSD elements into the symmetric lattice structure around each mass point neighboring eight voxels [15],[16]. In the structure, let us distribute mass points uniformly in a rheology object whose intervals are the same along X, Y, and Z axes (Fig.2(a)). Let N be the number of mass points and M_{object} be the total mass of rheology object. Therefore, each mass point is given by $M = M_{object}/N$. In the experiment, we use $M_{object} = 864$ [g], $N = N_x \times N_y \times N_z$ (N_x , N_y , N_z : the numbers of masses along X, Y, Z axes, respectively). For example, $M =$ about 1.02 for $N = 847 = 11 \times 7 \times 11$, or $M =$ about 0.15 for $N = 5733 = 21 \times 13 \times 21$.

In our experiment, a real rheology object is built by mixing wheat flour and water. The rheology object whose scales are $10\text{cm} \times 6\text{cm} \times 10\text{cm}$ along X, Y and Z axes, which are horizontally and vertically two times larger than its virtual rheology object whose scales are $5\text{cm} \times 3\text{cm} \times 5\text{cm}$ along the axes. Therefore, the former volume (600 [cm^3] = water weight [g]) is eight times larger than the latter volume (75 [cm^3]). Based on this similarity, force propagation and shape deformation of real object equals to these of virtual one.

The elements are inserted between neighboring mass points as illustrated in Fig.3(a). A virtual rheology object is deformed by expanding and contracting basic elements. Let $P_{i,j,k}$ be position vector corresponding to mass point (i, j, k) (i, j, k are integers, $1 \leq i \leq N_x$, $1 \leq j \leq N_y$ and $1 \leq k \leq N_z$). Let us derive quadratic differential equation of each mass at $P_{i,j,k}$. Each internal force acting on $P_{i,j,k}$ by the element between $P_{i,j,k}$ and its neighbor $P_{i+\alpha, j+\beta, k+\gamma}$ is denoted by $F_{i,j,k}^{\alpha, \beta, \gamma}$. For each mass, 6 shorter elements whose distance is denoted as l , 12 normal elements whose distance is denoted as $\sqrt{2}l$, and 8 longer elements whose distance is denoted as $\sqrt{3}l$ are located. The unit length l is defined as $5/(N_x - 1)$ cm for a virtual rheology object.

Therefore, a total internal force $F_{i,j,k}^{int}$ acting on $P_{i,j,k}$ is given by the sum of $26 (= 6 + 12 + 8)$ internal forces

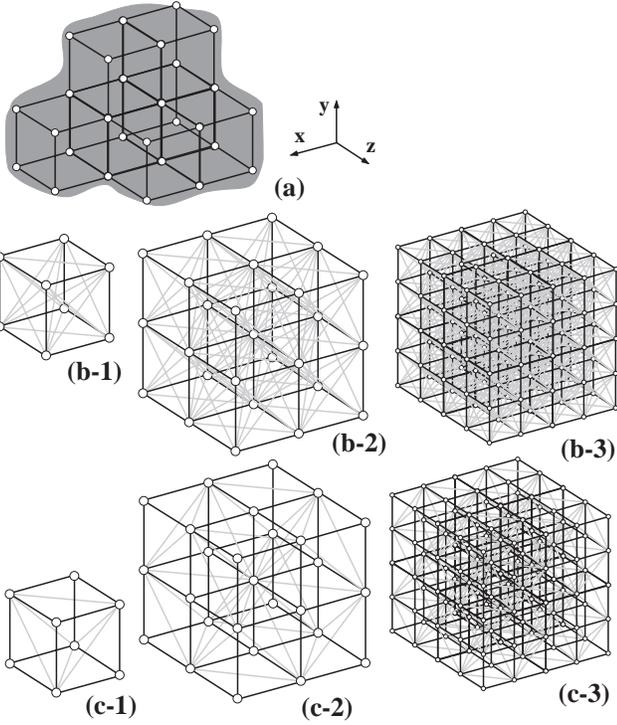


Fig. 2. (a) A real 3-D rheology object and its set of voxels. (b) The lattice structure in the voxel model. (c) The truss structure in the voxel model. (b-1),(c-1) The largest voxel. (b-2),(c-2) Smaller voxels whose mass number is $3 \times 3 \times 3$ and also voxel number is $2 \times 2 \times 2$. (b-3),(c-3) The smallest voxels whose mass number is $5 \times 5 \times 5$ and also voxel number is $4 \times 4 \times 4$.

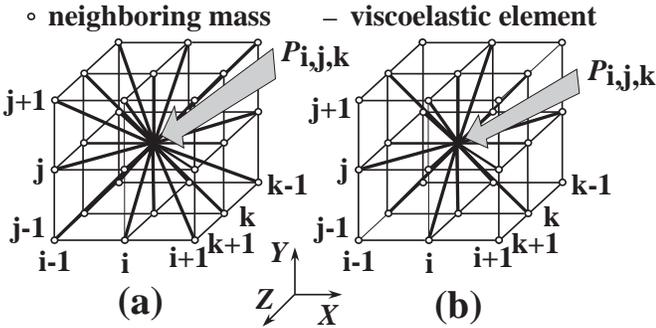


Fig. 3. (a) Neighboring mass points by basic elements in the lattice structure. (b) Neighboring mass points by basic elements in the truss structure.

$F_{i,j,k}^{\alpha,\beta,\gamma}$. Moreover, if the sum of active external forces at $P_{i,j,k}$ is denoted by $F_{i,j,k}^{ext}$, we obtain the following quadratic differential equation. This summation requires small calculation time.

$$M\ddot{P}_{i,j,k} = F_{i,j,k}^{int} + F_{i,j,k}^{ext} \quad (1)$$

In order to calculate a next position $P_{i,j,k}$ ($1 \leq i \leq N_x$, $1 \leq j \leq N_y$ and $1 \leq k \leq N_z$) at each mass, we should solve the above differential equation. This is done by the relatively faster Euler method, but it is still expensive.

$$F_{i,j,k}^{int} = \sum_{\substack{\alpha,\beta,\gamma \in \{-1,0,1\} \\ (\alpha,\beta,\gamma) \neq (0,0,0)}} F_{i,j,k}^{\alpha,\beta,\gamma} \quad (2)$$

Note that mass positions are fixed on surfaces of rheology object pushed by a rigid body and a whole floor. This means the integrations by the Euler method can be neglected.

C. A Basic Truss (Tetrahedral) Structure

As mentioned previously, a real 3-D rheology object is expressed by a set of small voxels as its virtual one (Fig.2(a)). Then in the last paragraph, we allocate many 1-D MSD elements by the lattice structure in the voxel model (Fig.2(b)). In this paragraph, we allocate many 1-D MSD elements by the truss structure where many tetrahedrons exist in the voxel model (Fig.2(c)). The difference between lattice and truss structures is briefly explained as follows: In the lattice structure, each mass point always connects to all the neighbor points whose distances are l , $\sqrt{2}l$ and $\sqrt{3}l$ (Fig.3(b)). On the other hand, in the truss model, each voxel includes four smaller tetrahedrons and one larger tetrahedron. They have many edges whose lengths are l and $\sqrt{2}l$. Consequently, all longest pairs are eliminated, whose distance is $\sqrt{3}l$, and also longer pairs are partially eliminated, whose distance is $\sqrt{2}l$ (Fig.3(b)).

The symmetries of force propagation and shape deformation are always maintained in the lattice structure (Fig.2(b-1),(b-2),(b-3)). As contrasted with this, any symmetry of force propagation or shape deformation is not kept at all in the truss structure (Fig.2(c-1)). To overcome this drawback, we pair two voxels whose truss structures are opposite (Fig.2(c-2),(c-3)), For this reason, in order to construct symmetric voxel/truss structure, we prepare a set of voxels whose N_x , N_y and N_z are set as odd numbers.

In the voxel/lattice model, we regard an arbitrary mass point as $P_{i,j,k}$ ($1 \leq i \leq N_x$, $1 \leq j \leq N_y$, $1 \leq k \leq N_z$), its neighbor mass points are denoted as $P_{i+\alpha,j+\beta,k+\gamma}$ ($\alpha, \beta, \gamma \in \{-1,0,1\}, (\alpha, \beta, \gamma) \neq (0,0,0)$). Therefore, internal forces between the neighbor masses are represented as $F_{i,j,k}^{\alpha,\beta,\gamma}$ under two equations (1) and (2). On the contrary, in the voxel/truss model, since each voxel includes five tetrahedrons, the number of pairs of neighbors decreases. For example, if $i+j+k$ is odd at an arbitrary point $P_{i,j,k}$ (three i, j and k are completely odd or the two are even and one is odd), α, β and γ are satisfied under the following condition:

$$|\alpha| + |\beta| + |\gamma| = 1 \quad (3)$$

For this reason, there are six pairs of neighbors whose length is l (Fig.4(a)). In addition, if $i+j+k$ is even at the mass point $P_{i,j,k}$ (three i, j and k are completely even or the two are odd and one is even), α, β and γ are satisfied under the following condition:

$$|\alpha| + |\beta| + |\gamma| \neq 3 \quad (4)$$

For this reason, there are two kinds of MSD elements whose lengths are shorter l and longer $\sqrt{2}l$ in the voxel/truss model (Fig.4(b)). Compared with the voxel/lattice model, there is no element whose length is the longest $\sqrt{3}l$ and also there is a few elements whose length is the longer $\sqrt{2}l$ in the voxel/truss model described in Table I.

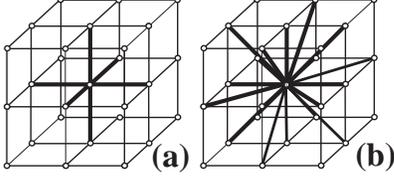


Fig. 4. (a) A set of MSD elements satisfied in the equation (3). (b) The other set of these satisfied in the equation (4).

TABLE I

A COMPARISON OF NUMBERS OF MASS POINTS AND BASIC ELEMENTS BETWEEN THE LATTICE AND TRUSS STRUCTURES.

11 × 7 × 11 resolution		
	number of masses	number of elements
lattice	847	8706
truss	847	4286
21 × 13 × 21 resolution		
lattice	5733	65972
truss	5733	31492

D. Feed-Forward (Local) Volume Constant Condition in the Lattice Structure

The local volume constant condition extends a voxel during deformation by eliminating four longest elements (whose distances are $\sqrt{3}l$ cm in Fig.5(a)) and adding eight half-length elements from its center of gravity to eight vertices (whose distances are $\sqrt{3}l/2$ cm in Fig.5(b)).

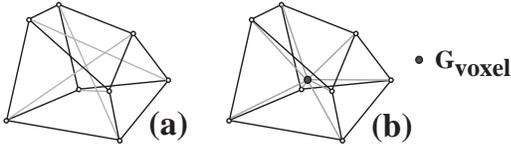


Fig. 5. Local volume constant condition: (a) Four longest elements directly connecting two opposite vertices in a voxel. (b) Eight shorter elements connecting from the center G_{voxel} of gravity of a voxel to its eight vertices.

This technique has been already used in a MS (mass-spring) model [21]. This controls the isotropy or anisotropy of elastic material. This idea can be straightforwardly extended to rheologic material controlled by a MSD (mass-spring-damper) model. Since this needs MSD elements whose lengths are $\sqrt{3}l$, this can be applied for not the voxel/truss model but the voxel/lattice model.

E. Feed-back (Global) Volume Constant Condition in the Truss Structure

The local volume constant condition is a feed-forward approach and thus cannot decrease the difference between

real and virtual objects directly. For this reason, the volume of virtual object does not equal to that of real one. Also, the local volume constant condition does not ensure stability of shape deformation, especially if all uncertain parameters are not well calibrated. To overcome this, we test another volume constant condition which is similar to [16],[22],[23].

According to the global volume constant condition based on the **Pascal's Principle**, we always add external forces at all mass points around the surface of virtual rheology object by minimizing the volume difference between V_{int} and V (V_{int} : the volume of real rheology object, and V : the volume of virtual rheology object). If $V \leq V_{int}$ is satisfied, a negative unit magnitude p of external force appears around a virtual rheology object (Fig.6(a)), otherwise, a positive unit magnitude p of external force appears around the object (Fig.6(b)) (\dot{V} : time difference of the volume V , K_{vol} : elastic coefficient of global volume constant condition, C_{vol} : viscous coefficient of global volume constant condition).

$$p = -K_{vol}(V - V_{int}) - C_{vol}\dot{V} \quad (5)$$

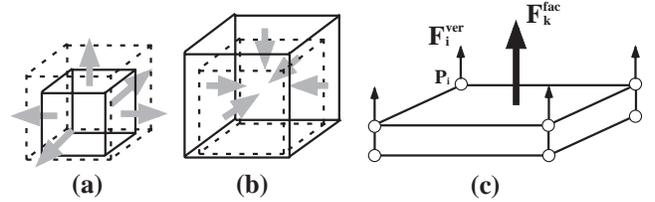


Fig. 6. Global volume constant condition: (a) If $V \leq V_{int}$ is satisfied, an outward external force appears around a virtual rheology object. (b) If $V_{int} \leq V$ is kept, an inward external force occurs around it. (c) An external force F_k^{fac} on a patch P_k is distributed into four forces F_i^{ver} of its corners.

Furthermore, concerning to a patch P_k whose number is k , area is S_k , and outward unit normal vector is n_k^{out} . By the global volume constant condition, there is an additional force F_k^{fac} at a patch P_k (Fig.6(c)).

The force F_k^{fac} acting to the patch P_k is denoted as the following equation.

$$F_k^{fac} = pS_k n_k^{out} \quad (6)$$

In succession, the force F_k^{fac} is distributed into four forces at corner masses of P_k . Therefore, the force F_i^{ver} acting to each corner mass is denoted by the following equation (i : mass number around a virtual rheology object, A_i : a set of all patches including the mass).

$$F_i^{ver} = \sum_{k \in A_i} \frac{F_k^{fac}}{4} \quad (7)$$

The total sum of all external forces at all mass points around a virtual rheology object amounts to zero. This means that the set of external forces around the virtual rheology object is always zero. This means that the set of internal forces is always constant and consequently momentum variation is completely fixed. Furthermore, as long

as two coefficients K_{vol} and C_{vol} are set large enough, the volume of virtual rheology object nearly equals to that of real rheology object stably. These parameters should not be calibrated at all because of the feedback property. Finally, the global volume constant condition is categorized into a feedback method for decreasing the volume difference between virtual and real objects. Therefore using the global volume constant condition, the volume of virtual object is precisely coincident with that of real one.

III. CALIBRATE MANY UNCERTAIN PARAMETERS

The defective point of MSD is accuracy of force propagation and shape deformation. To overcome this, we calibrate many uncertain parameters of four models by minimizing the difference between shape deformations of real and virtual rheology objects. For this purpose, we calibrate 16 coefficients of spring, dampers, and ratio between Voigt and damper parts of our MSD elements in the voxel/lattice model with the local volume constant condition, and do 8 ones in the other three models by a genetic algorithm (GA).

A. How to Calculate Shape Difference Between Real and Virtual Rheology Objects

In this research, a rheology object is precisely pushed by a rigid body located at the tip of a robotic manipulator (Fig.7(a)). The deformation, that is, the sequence of shapes is measured by two stereo vision camera systems *Digiclops* and its software development kit (SDK) *Triclops* (provided by Point Grey Research Inc, Canada). Each captures about three or more thousand points as shape of real rheology object in the real-time manner. After capturing the shape deformation, we finally measure how much total volume is changed before pushing and after releasing the rheology object. For this purpose, we use the following primitive method. First of all, we fill a ball with water, and then drop a deformed object into the ball (Fig.7(b)). Secondly, we gather overflowed water and measure its weight by a precise electric balance. As a result, we can understand volume of rheology object always decreases by about 3 percentages.



Fig. 7. (a) An experiment system: A real rheology object is pushed by a rectangular rigid body located at the tip of a robotic manipulator, and deformations of its sides are measured by two *Digiclops* cameras. (b) The volume difference between an initial real rheology object and its pushed one is measured by the overflowed water.

In order to evaluate the difference between real and virtual rheology objects, we summarize minimum distances from captured points to their nearest surfaces around a virtual rheology object. The object consists of $N_x \times N_y \times N_z$

hexahedrons which are individually deformed from initial cubes. Therefore, in order to evaluate the difference, we calculate the minimum of $N_x \times N_y \times N_z$ shortest distances for each captured point to all hexahedrons by the Lin-Canny closest point algorithm [24], and then we calculate the sum S of all the minimum distances for all captured points, which are larger than the average error $0.05cm$ of *Digiclops*. In our calibration, we use the sum of four S at four times during and after each pushing operation (Fig.8).

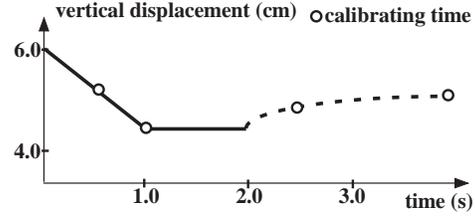


Fig. 8. The vertical displacement along the Y-axis in the pushing, keeping and releasing operations described as black whole and dot lines.

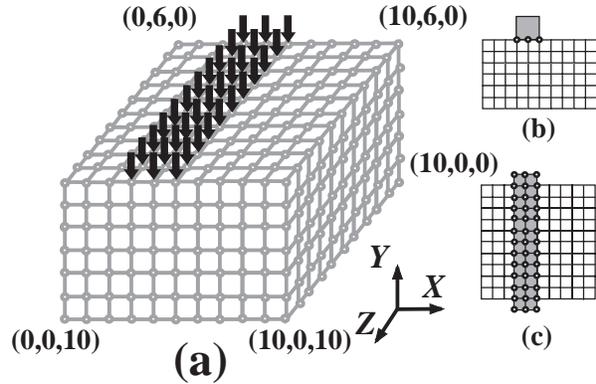


Fig. 9. (a) 3-D view for the pushing operation. (b) Front view. (c) Upper view.

By minimizing the total S in a genetic algorithm, we can obtain a better set of at most 16 (at least 8) C_1 , C_2 , K and ratio a .

B. Genetic Algorithm (GA)

1. Initialize four parameters G_{ind} , G_{gen} , G_{sur} , G_{mut} . G_{ind} is the number of individuals, G_{gen} is a generation threshold. G_{sur} is a survival ratio of individuals from last generations to present ones ($0 \leq G_{sur} \leq 1$). G_{mut} is a mutation ratio ($0 \leq G_{mut} \leq 1$). In addition, set present generation P_{gen} as 0.

2. Generate individuals whose number is G_{ind} . Each consists of calibrating parameters such as K , C_1 and so on. The parameters are randomly selected within their intervals.

3. If P_{gen} amounts to G_{gen} , this algorithm finishes. Otherwise, evaluate and sort all individuals by shape differences S_n ($n = 1, 2, \dots, G_{ind}$) between real and virtual objects for all individuals.

4. Select better individuals whose number is $G_{ind} \times G_{sur}$. We call this as selection.

5. Generate a few individuals by reversing bits of their original ones, whose number is $G_{ind} \times G_{mut}$. This operation is called as mutation.

6. Select two individuals (A and B). Each individual is bit-sized, and its crossing point is randomly selected. Then, cut each at the crossing point to generate former and latter parts (A_f and A_l , or B_f and B_l). Then, one former and another latter are combined ($A_f + B_l$) to make the other individuals. This is called as crossing. Finally, after incrementing P_{gen} , we return to the step 3.

Finally, $G_{ind} \times (1 - G_{sur})$ individuals newly appear in the next generation by crossing or mutation ($G_{ind} \times G_{sur}$ individuals are left from the last generation).

IV. COMPARATIVE RESULTS

In this section, we compare four structures with each other concerning to computation time, memory storage and shape accuracy. The deformation of virtual rheology object is calculated and visualized by a 3-D graphics software OpenGL in a personal computer (CPU: Pentium4 3.00GHz, Main memory: 2048MB) with a 3-D graphics acceleration board (GeForce FX 5600, 128MB).

In all structures, we distinguish MSD elements in surface and core areas of virtual rheology object to calibrate (Fig.10). The first set is K^{surf} , C_1^{surf} , C_2^{surf} , a^{surf} within the surface area, and the second set is K^{core} , C_1^{core} , C_2^{core} , a^{core} within the core area. Furthermore, in the lattice structure with the local volume constant condition, we distinguish MSD elements on and in each voxel. That is, we categorize shortest 8 diagonal elements illustrated in Fig.5 as basic elements in a voxel, and categorize the other longer 18 elements as basic elements on a voxel. Therefore, the first set is $K^{surf-in}$, $C_1^{surf-in}$, $C_2^{surf-in}$, $a^{surf-in}$ in each voxel within surface area, the second set is $K^{core-in}$, $C_1^{core-in}$, $C_2^{core-in}$, $a^{core-in}$ in each voxel within core area, the third set is $K^{surf-on}$, $C_1^{surf-on}$, $C_2^{surf-on}$, $a^{surf-on}$ on each voxel within surface area, and the fourth set is $K^{core-on}$, $C_1^{core-on}$, $C_2^{core-on}$, $a^{core-on}$ on each voxel within core area. Finally, so as to form a virtual object pushed by a rigid body stably (Fig.11(a)), we give the following intervals $50 \leq K^{surf-in}, K^{core-in} \leq 3000$, $100 \leq K^{surf-on}, K^{core-on} \leq 3000$, $250 \leq C_1^{surf-in}, C_1^{core-in} \leq 10000$, $500 \leq C_1^{surf-on}, C_1^{core-on} \leq 10000$, $250 \leq C_2^{surf-in}, C_2^{core-in} \leq 20000$, $500 \leq C_2^{surf-on}, C_2^{core-on} \leq 20000$, $0.3 \leq a^{surf-in}, a^{core-in}, a^{surf-on}, a^{core-on} \leq 0.7$. If some of them are too large, each element becomes unstable and consequently shape of virtual rheology object is crushed as illustrated in Fig.11(b).

A. Computation Complexity and Memory Storage

In the virtual 3-D graphics world, we construct the lattice and truss structures without any volume constant condition, the lattice structure with the local one, and the truss structure with the global one. First of all, we generate 2000 deformations of virtual rheology object in simulation, which correspond to their deformations of real rheology object in all experimental trials by 4 [s]. Therefore, each

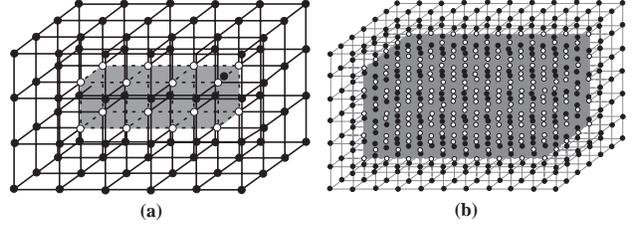


Fig. 10. Surface and core areas of virtual rheology object. Black and white mass points are located on the surface and core areas, respectively. Gray voxel and dotted line are among the core area. (a) $5 \times 3 \times 3$ model. (b) $10 \times 6 \times 6$ model.

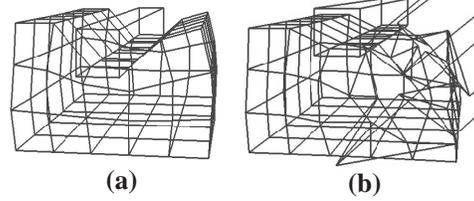


Fig. 11. (a) A stable shape of a rheology object. (b) An unstable shape of the object.

deformation time in simulation corresponds to 2 [ms] in experiment. Secondly in each simulation, we make their deformations of virtual rheology object by about 60 [s] in the lattice structure without any condition or 50 [s] in the truss structure without it or 60 [s] in the lattice structure with the local volume constant condition or 90 [s] in the truss structure with the global one, whose size is $11 \times 7 \times 11$. Therefore, PC and graphics acceleration board require about 30 or 25 or 30 or 45 [ms] per each deformation (Table II).

TABLE II
CALCULATION TIME [msec] AND MEMORY STORAGE [MB] PER ONE DEFORMATION FOR THE PUSHING OPERATION BY OUR PC.

	Total calculation time [msec]	
	[$11 \times 7 \times 11$]	[$21 \times 13 \times 21$]
lattice str. wo. any cond.	30.727	229.094
truss str. wo. any cond.	24.039	172.758
lattice str. wi. local cond.	31.109	230.164
truss str. wi. global cond.	46.633	347.157
Deformation calculation time [msec]		
lattice str. wo. any cond.	15.033	122.182
truss str. wo. any cond.	8.775	61.103
lattice str. wi. local cond.	16.093	128.683
truss str. wi. global cond.	31.239	244.315
Memory storage [MB]		
lattice str. wo. any cond.	44.648	99.988
truss str. wo. any cond.	44.808	100.720
lattice str. wi. local cond.	45.032	100.964
truss str. wi. global cond.	44.476	100.684

Total calculation time T_{cal} in Table II is the sum of deformation time T_{def} and the other time T_{ove} ($T_{cal} = T_{def} + T_{ove}$). T_{ove} is mainly the time to calculate S_n , which proportionally depends on the number of voxels, i.e., $(N_x - 1) \times (N_y - 1) \times (N_z - 1)$. In the lattice and truss structures without any volume constant condition, the time T_{def} for calculating deformation almost equals to time T_{int} for integrating quadratic differential equations at many mass points, i.e., $T_{def} = T_{int}$. T_{int} in four models directly depends on the number of our MSD elements, i.e.,

$(N_x - 1) \times (N_y - 1) \times (N_z - 1)$. Furthermore, in the truss structure with the global one, we should additionally consider time T_{vol} to calculate the total volume of virtual rheology object by summing up volumes of all polyhedrons deformed from voxels. Also, T_{vol} directly depends on $(N_x - 1) \times (N_y - 1) \times (N_z - 1)$. For this reason, only in the truss structure with the global condition, time T_{def} to calculate each deformation equals to total time T_{int} and T_{vol} , i.e., $T_{def} = T_{int} + T_{vol}$. Since numbers of mass points in four kinds of models are almost the same, memory storage is also the same. The memory storage m is approximately evaluated by summing m_N and m_S . The mass storage m_N directly depends on the number of masses, and the software storage m_S is determined as the sum of C++ compiler (VC++ 6.0) and OpenGL software. It is invariable. From Table II, we understand $m_S = 33[MB]$ (C++ compiler = $20[MB]$ and OpenGL = $13[MB]$) and $m_N = 12 \times N[KB]$ ($N = N_x \times N_y \times N_z$: the number of mass points).

B. Deformation Accuracy

In this paragraph, we calibrate a better set of uncertain parameters by a genetic algorithm (GA). Firstly, we set G_{ind} as a small value 50. The reason is that similar individuals frequently appear even though G_{ind} is larger. Then, we set G_{sur} as a small value 0.2 because of the same reason. Thirdly, we set G_{gen} as 250 (the lattice structure without any volume constant condition) or 360 (the truss structure without any volume constant condition) or 250 (the lattice structure with the local volume constant condition) or 170 (the truss structure with the global volume constant condition) in $11 \times 7 \times 11$ resolution. After determining $G_{ind} = 50$ and $G_{sur} = 0.2$, we always get 40 individuals in each generation. Therefore, in order to get 10080, 12100, 10080 and 6720 individuals, we pass through 250, 300, 250 and 170 generations, respectively. Thus, calibrating time by GA turns into 168 hours. Finally, since G_{mut} should be selected as a smaller value, we set G_{mut} as 0.01.

All calibrated parameters, error distance and volume precision between real and virtual rheology objects are described for the lattice and truss structures without any volume constant condition, the lattice structure with the local one, and the truss structure with the global one, whose size is $11 \times 7 \times 11$ in Table III. In addition, shape deformation after releasing is illustrated in Fig.12 for the lattice and truss structures without any volume constant condition, the lattice structure with the local one, and the truss structure with the global one, whose size is $11 \times 7 \times 11$. Every pushing operation is illustrated in Fig.8 and 9.

If a virtual rheology object modeled in the voxel/lattice and voxel/truss models without any volume constant condition is pushed by the operation, expanding the rheology object is not enough (the volume of virtual rheology object is extremely smaller than that of its real one. The real volume is $72.75 [cm^3] = 75 [cm^3] \times 0.97$). This means that force propagation of a real rheology object cannot be realized by these models themselves. This tendency is

TABLE III
CALIBRATION RESULTS BY GENETIC ALGORITHM FOR THE PUSHING OPERATION IN FOUR STRUCTURES.

The number of captured points is $N = 15372$				
Calibration result	lattice struct. without local v.c.c	truss struct. without global v.c.c	lattice struct. with local v.c.c	truss struct. with global v.c.c
$S [cm]$	2201	2583	1506	1908
$V [cm^3]$	70.56	69.28	72.88	74.83
$K^{core} [gf/cm^3]$	2946	177	2424	3000
$C_1^{core} [gfs/cm^3]$	9992	8653	4901	500
$C_2^{core} [gfs/cm^3]$	19598	5395	2664	14798
a^{core}	0.69	0.34	0.61	0.70
$K^{surf} [gf/cm^3]$	2968	2717	1899	2997
$C_1^{surf} [gfs/cm^3]$	768	519	5837	2941
$C_2^{surf} [gfs/cm^3]$	19883	582	10606	19998
a^{surf}	0.69	0.68	0.47	0.70
	$K^{-in} [gf/cm^3]$	$C_1^{-in} [gfs/cm^3]$	$C_2^{-in} [gfs/cm^3]$	a^{-in}
lattice st. wi. l.v.c.c - core	1902	2015	1759	0.31
lattice st. wi. l.v.c.c - surf	236	603	19003	0.59
The number of captured points whose errors are more than 0.25cm				
	first	second	third	total
lattice st. wo. l.v.c.c	700	623	584	2454
truss st. wo. g.v.c.c	1076	1043	785	3704
lattice st. wi. l.v.c.c	219	167	232	885
truss st. wi. g.v.c.c	668	439	343	1819

remarkable especially in the voxel/truss model because its number of MSD elements is smaller than the number of them in the voxel/lattice model. To overcome this defective point, we add local and global volume constant conditions into the voxel/lattice and voxel/truss models, respectively. The reason is as follows: the local (feed-forward) condition cannot be applied for the voxel/truss model and also the global (feed-back) condition is about two times slower than the local (feed-forward) one. The feed-back condition strictly leads the volume of a virtual rheology object to that of a real one. For this reason, the volume of voxel/truss model with the feed-back condition always converges to that of a real rheology object (Table III). However, the shape of voxel/truss model with the feed-back condition is not always coincident to that of a real rheology object (Fig.12 and Table III). This perhaps occurs by eliminating four MSD elements whose lengths are the longest. This problem can be solved in the lattice structure with the local volume constant condition because it reasonably expands each voxel to get shape of a rheology object which is quite similar to shape of a real one.

V. CONCLUSIONS

In this paper, we compared lattice and truss structures in the same voxel MSD model. The lattice structure excludes and includes the local volume constant condition, and also the truss structure excludes and includes the global volume constant condition. The concluding remarks are as follows:

- (1) The smaller order of differences between volumes and shapes of virtual and real rheology objects during the deformation is as follows: (lattice structure with the local volume constant condition) \gg (truss structure with the global volume constant condition) \gg (lattice structure without the local volume constant condition) \gg (truss structure without the global volume constant condition).

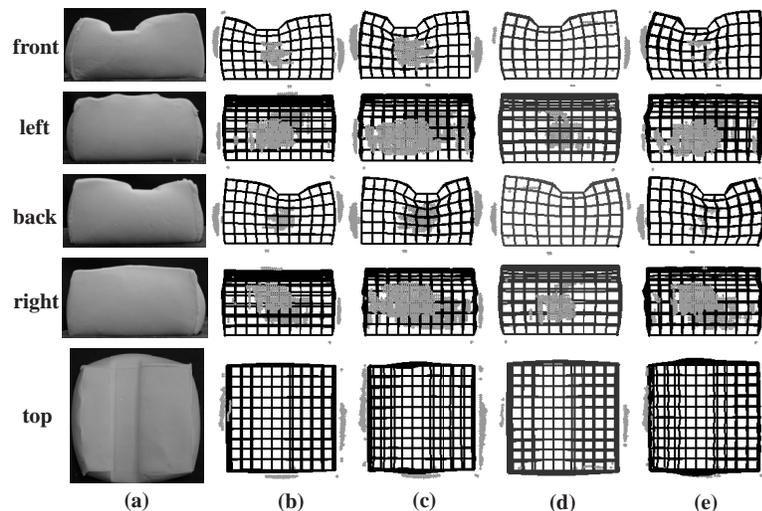


Fig. 12. Deformation differences between real and virtual rheology objects for the pushing operation. (a) Real rheology object. (b),(c),(d),(e) Virtual rheology objects. Areas are colored by dark gray, whose errors are larger than 0.25 [cm] in the lattice and truss structures without any volume constant condition, the lattice structure with the local one, and the truss structure with the global one.

(2) The smaller order of calculation costs is as follows: (truss structure without the global volume constant condition) > (lattice structure without the local volume constant condition) > (lattice structure with the local volume constant condition) \gg (truss structure with the global volume constant condition).

(3) In consequence, (lattice structure with the local volume constant condition) is the best structure in the real-time dynamic animation.

In future, we try to compare another structure and/or basic MSD element with the proposed ones

ACKNOWLEDGMENT

The authors thank Prof. H.Tanaka (Ritsumeikan Univ.), H.Fujii (Osaka Electro Commun. Univ.) for discussions and C programming, respectively. This is supported in part by 2004 Grants-in-aid for Scientific Research from the Ministry of Education, Science and Culture, Japan (No.14550247).

REFERENCES

- [1] Y.Zhuang, "Real-time simulation of physically-realistic global deformations," *Doctoral Thesis, UC Berkeley*, 2000.
- [2] G.Debunne, M.Desbrun, M.Paule Cani and A.Barr, "Dynamic real-time deformations using space and time adaptive sampling," *Computer Graphics (Proc. SIGGRAPH)*, pp.31-36, 2001.
- [3] X.Wu, M.S.Downes, T.Goktekin and F.Tendick, "Adaptive nonlinear finite elements for deformable body simulation using dynamic progressive meshes", *Computer Graphics Forum (Proc. Eurographics)*, pp.349-358, 2001.
- [4] K.Waters, "A muscle model for animating three-dimensional facial expression," *Computer Graphics (Proc. SHIGGRAPH)*, pp.17-24, 1987.
- [5] E.Promayon, P.Baconnier and C.Puech, "Physically-based deformations constrained in displacements and volume," *Computer Graphics Forum (Proc. Eurographics)*, pp.155-164, 1996.
- [6] J.Louchet, X.Provot and D.Crochemore, "Evolutionary identification of cloth animation models", *Computer Animation and Simulation (Eurographics)*, pp 44-54, 1995.
- [7] A.Joukhadar, F.Garat and C.Laugier, "Parameter identification for dynamic simulation", *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp.1928-1933, 1997.
- [8] D.Terzopoulos, J.Platt, A.Barr and K.Fleischer, "Elastically deformable models," *Computer Graphics (Proc. SIGGRAPH)*, pp.205-214, 1987.
- [9] D.James and D.K.Pai, "ArtDefo, accurate real time deformable objects," *Computer Graphics (Proc. SIGGRAPH)*, pp.65-72, 1999.
- [10] M.B.Nielsen and S.Cotin, "Real-time volumetric deformable models for surgery simulation using finite elements and condensation," *Computer Graphics Forum (Proc. Eurographics)*, pp.57-66, 1996.
- [11] M.Muller, J.Dorsey, L.McMillan, R.Jagnow and B.Cutler, "Stable real-time deformations", *Proceedings of ACM SIGGRAPH Symposium on Computer Animation (SCA)*, pp 49-54, 2002.
- [12] J.J.Berkley, "Haptic rendering of deformable bodies using real-time finite element analysis: an application to surgical simulation," *Doctoral Thesis, University of Washington*, 2002.
- [13] S.P.DiMaio, "Modelling, simulation and planning of needle motion in soft tissues", *Doctoral Thesis, University of British Columbia*, 2003.
- [14] H.-W. Nienhuys, "Cutting in deformable objects", *Doctoral Thesis, Utrecht University*, 2003.
- [15] S.Tokumoto, Y.Fujita and S.Hirai, "Deformation modeling of viscoelastic objects for their shape control," *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp.1050-1057, 1999.
- [16] S.Tokumoto, S.Hirai, and H.Tanaka, "Constructing virtual rheological objects," *Proc. World Multiconference on Systemics, Cybernetics and Informatics*, pp.106-111, July, Auland, 2001.
- [17] M.Kimura, Y.Sugiyama, S.Tomokuni, and S.Hirai, "Constructing rheologically deformable virtual objects," *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp.3737-3743, 2003.
- [18] H.Noborio, R.Enoki, S.Nishimoto and T.Tanemura, "On the calibration of deformation model of rheology object by a modified randomized algorithm," *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp.3729-3736, 2003.
- [19] H.Noborio, R.Nogami and R.Enoki, "Precise deformation of rheology MSD model calibrated by randomized algorithm," *Eurographics 2003 (Short Presentations)*, pp.171-178, 2003.
- [20] S.Tomokuni, Y.Sugiyama and S.Hirai, "Construction of virtual rheological objects computable in realtime," *Trans. of Virtual Reality Society of Japan*, Vol.8, No.3, pp.247-254, 2003 (in Japanese).
- [21] D.Bourguignon and M.-P.Cani, "Controlling anisotropy in mass-spring systems," *Proc. of the 11th Eurographics Workshop on Computer Animation and Simulation*, pp.113-123, 2000.
- [22] I.Costa and R.Balaniuk, "Static solution for realtime deformable objects with fluid inside," *In ERCIM News*, pp.44-45, 2001.
- [23] K.Sundaraj, C.Laugier, and I.F.Costa, "An approach to LEM modelling : construction, collision detection and dynamic simulation," *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp.2196-2201, 2001.
- [24] M.C.Lin and J.F.Canny, "A fast algorithm for incremental distance calculation," *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp.1008-1014, 1991.

第4章 連続体力学に基づく物体変形の パーティクルベースモデリング

4.1 緒言

仮想レオロジー物体における研究課題として、大変形シミュレーションが挙げられる。食品加工においては、小麦粉生地大きな変形が加わる場面が多く見られる。たとえば、麺体を練るという加工工程において、麺体は元の厚さに比して10%の厚さまで伸ばされる場合がある。そのため、小麦粉生地の変形シミュレーションにおいて、大変形に対応したいというニーズが存在する。しかし、物体変形シミュレーションにおいて大変形は困難な課題であり、現状においては大変形時のシミュレーションの安定性に難がある。本章では、安定に大変形シミュレーションが可能な物体変形モデルを提案する。

一般に三次元物体の変形モデリングは、伸張変形、剪断変形、体積変形の3つの観点から行われる。しかし、力学的等方性を持つ物体において、この3つは独立ではなく、この内の二つをモデリングすれば、物体の変形特性を表現することができる。そこで、本章では、剪断モデルおよび体積モデルの二つをモデリングする。大変形に対応し、安定にリアルタイム変形シミュレーションが可能かつ連続体モデルのパラメータが適用可能なレオロジー物体変形モデルの構築を目指す。

4.2 剪断モデル

4.2.1 二次元物体における剪断モデル

運動量保存則と角運動量保存則の両者を満たす剪断モデルを提案する。まず、長方形の物体が図4.1に示す形状に変形した場合、図4.2-(a)方向の剪断に対する剪断ひずみ γ_a は、

$$\gamma_a = \frac{l_{ha}}{l_{va}} - \frac{l_{ha}^{init}}{l_{va}^{init}} \quad (4.1)$$

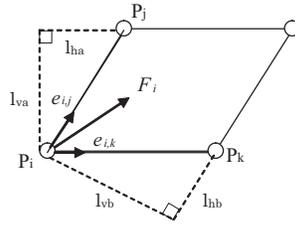


図 4.1: 質点にかかる合力

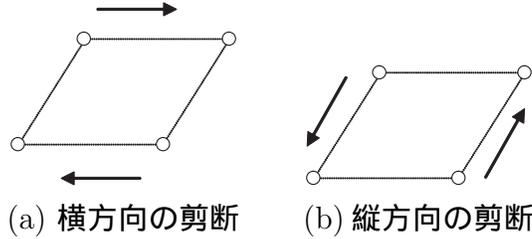


図 4.2: 剪断

である．ただし，ここで l_{ha} は稜線 P_iP_j における稜線 P_iP_k 方向の成分， l_{va} は稜線 P_iP_j における稜線 P_iP_k の垂直方向の成分であり， l_{ha}^{init} ， l_{va}^{init} はそれらの初期状態における値である．

剪断ひずみに対して発生する剪断応力はフォークトモデル，三要素モデル，一般化フォークトモデル等の任意の力学要素で表現できる．たとえば，力学モデルにフォークトモデルを選択した場合，剪断弾性率を G^{ela} ，剪断粘性率を G^{vis} で表すと，剪断応力は，

$$\sigma_a^{shear} = -G^{ela} \Delta\gamma_a - G^{vis} \dot{\gamma}_a \quad (4.2)$$

となる．また，図 4.2-(b) 方向における剪断ひずみおよび剪断応力は $\gamma_b = \gamma_a$ ， $\sigma_b = \sigma_a$ となる．これらより，図 4.1 における左側面の稜線 P_iP_j の長さを $l_{i,j}$ ，稜線 P_iP_j の単位ベクトルを $e_{i,j}$ ，下側の稜線 P_iP_k の長さを $l_{i,k}$ ，単位ベクトルを $e_{i,k}$ とすると，左下の質点 P_i にかかる剪断力のベクトルは，

$$\mathbf{F}_i^{shear} = \frac{1}{2} \sigma_a^{shear} l_{i,j} e_{i,j} + \frac{1}{2} \sigma_b^{shear} l_{i,k} e_{i,k} \quad (4.3)$$

となる．

剪断要素において，図 4.2-(a) に示されるような上辺および下辺にかかる単一方向の剪断力だけでは角運動量保存則は満たされず，図 4.2-(b) に示されるような側面の辺に発生する力も含めた全体の合力において角運動量保存則が満たされる．そこで，角運動量保存則を導入するために，図 4.1 に示すように，質点にかかる合力に着目

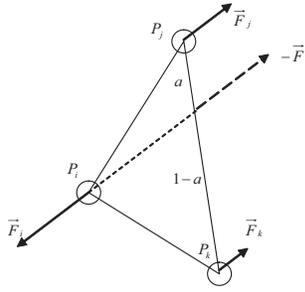


図 4.3: エッジにかかる力の分配

する．ここで，左下の質点 P_i およびそれに繋がる質点 P_j, P_k からなる計三つの質点において，運動量保存則と角運動量保存則が満たされるような反作用を仮定する．質点系において直線上にある二つの質点に対して，同じ大きさの力が逆向きにかかる時，運動量保存則と角運動量保存則が同時に満たされることが知られている．同様に，図 4.3 に示すように，質点 P_i を通り，剪断力ベクトル F_i^{shear} と同一の向きを持つ直線と，稜線 $P_j P_k$ の交点に， $-F_i^{\text{shear}}$ を加えることで運動量保存則と角運動量保存則が満たされる．稜線は質量を持たないため，稜線に加わる力は両端の質点に分配される．これは，前述の交点の稜線上での位置に基づいて，分配する事ができる．図 4.3 に示すような交点の稜線における比率を a_{cf} とすると，質点 P_j ，質点 P_k にかかる力は，それぞれ

$$F_j^{\text{cf}} = -(1 - a_{cf}) F_i^{\text{shear}}, \quad (4.4)$$

$$F_k^{\text{cf}} = -a_{cf} F_i^{\text{shear}} \quad (4.5)$$

となる．

この剪断力学要素を図 4.4 に示すように，4つの質点全てに適用する．すなわち，図 4.4 において黒い丸で示された質点が (4.3) 式によって計算される力を適用する対象となる質点であり，白い丸で示された質点が (4.4), (4.5) 式による反力が加わる点である．菱形状物体においてこの4回の計算を行うと，各質点に生じる力は材料力学における基本理論と合致する値となる．これは，正方形の物体が菱形状に変形した場合，剪断力学要素によって生じる反作用が対角どうしでつりあい，打ち消しあうことにより，各質点に直接かけた剪断力のみが残るためである．

物体が菱形以外の形状に変形した場合では，剪断力のみでは運動量保存則および角運動量保存則が満たされなくなる可能性がある．しかし，本節で述べた反作用を加えることで，常に運動量保存則および角運動量保存則が満たされる．

本節で述べた剪断モデルは三角形にも適用することが可能である．その場合，作用点および反作用点は図 4.5 に示すものとなる．剪断力および反作用の式は四角形の

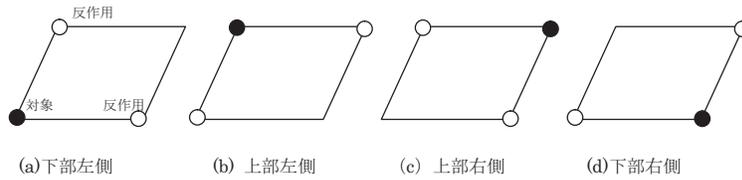


図 4.4: 剪断力の加わる作用点および反作用点

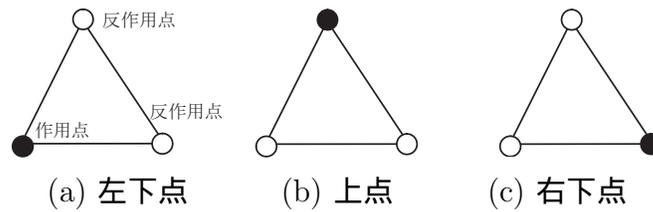


図 4.5: 三角形における剪断力の作用点および反作用点

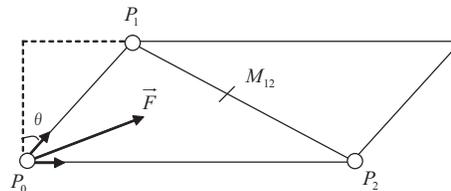


図 4.6: 平行四辺形状に変形した物体

ものと同一であり，(4.1)～(4.5)式を用いる．

4.2.2 二質点にかかる反作用の導出

この剪断モデルでは二質点にかかる反作用は常に等しい値となる．これを証明するため，剪断力の計算対象となる点 P_0 を原点とする図 4.6 の座標系において，反力のかかる位置を導出する．

剪断力の生じる質点を P_0 とし，反力の生じる質点を P_1, P_2 とする．また，質点 P_1 の位置を $[x_1 \ y_1]^T$ とし，質点 P_2 の位置を $[x_2 \ 0]^T$ とする．ここで，稜線 P_0P_1 の長さ $l_{0,1}$ および単位方向ベクトル $e_{0,1}$ は，それぞれ

$$l_{0,1} = \sqrt{x_1^2 + y_1^2} \quad (4.6)$$

$$\mathbf{e}_{0,1} = \frac{1}{l_{0,1}} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} \quad (4.7)$$

となり，線分 P_0P_2 の長さ $l_{0,2}$ および単位接線ベクトル $\mathbf{e}_{0,2}$ はそれぞれ

$$l_{0,2} = x_2 \quad (4.8)$$

$$\mathbf{e}_{0,2} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad (4.9)$$

となる．また P_1P_2 の中点 $M_{1,2}$ は

$$M_{1,2} = \frac{1}{2} \begin{bmatrix} x_1 + x_2 \\ y_1 \end{bmatrix} \quad (4.10)$$

となる．

点 P_0 に生じる剪断力 \mathbf{F}_0^{shear} の導出を行う．まず，エッジ P_0P_1 における剪断ひずみは，

$$\tan \theta = \frac{x_1}{y_1} \quad (4.11)$$

であり，また，エッジ P_0P_2 における剪断ひずみも同一の値となる．すなわち，図 4.6 において任意の四角形に対し，

$$\frac{l_{ha}}{l_{va}} = \frac{l_{hb}}{l_{vb}} = \tan \theta \quad (4.12)$$

が成り立つ．したがって，剪断弾性率を G とすると， P_0 に生じる剪断力 \mathbf{F}_0^{shear} は，

$$\mathbf{F}_0^{shear} = \frac{G}{2} \tan \theta (l_{0,1}\mathbf{e}_{0,1} + l_{0,2}\mathbf{e}_{0,2}) \quad (4.13)$$

となる．この剪断力に対する反力のかかる点は， \mathbf{F}_0^{shear} と同一の向きを持ち P_0 を通る直線と，直線 P_1P_2 の交点となる．この二つの直線それぞれにおける直線の式は，

$$y = \frac{y_1}{x_1 + x_2} x \quad (4.14)$$

$$y = \frac{y_1}{x_2 - x_1} (x_2 - x) \quad (4.15)$$

となる．(4.14),(4.15) 式より，交点 Q の座標は，

$$Q = \frac{1}{2} \begin{bmatrix} x_1 + x_2 \\ y_1 \end{bmatrix} \quad (4.16)$$

となる．(4.10) 式および (4.16) 式から，交点 Q は中点 $M_{1,2}$ と同一であり， P_1 および P_2 の中点となっていることが分かる．したがって，四角形の対角の変形形状が対称なものであれば双方の反力が拮抗し，打ち消されて 0 となる．すなわち，任意の長方形形状において，反力は 0 となる．また，任意の形状において反力は中点の位置となるため，射影等の計算を行う必要はなく，反力を $1/2$ ずつ二つの質点に割り振るだけでよい．

他の方法による反力の導出

反作用の計算式については別の方法で導出した場合にも同じ結果が得られる．反作用が P_1 および P_2 に割り振られる比率を r_1, r_2 とおくと，反作用 $\mathbf{F}_1^{\text{re}}, \mathbf{F}_2^{\text{re}}$ の定義は，

$$\begin{aligned}\mathbf{F}_1^{\text{re}} &= r_1 \mathbf{F}_0^{\text{shear}}, \\ \mathbf{F}_2^{\text{re}} &= r_2 \mathbf{F}_0^{\text{shear}}\end{aligned}$$

で代替できる．ここで，運動量保存則より，

$$\mathbf{F}_0^{\text{shear}} + r_1 \mathbf{F}_1^{\text{re}} + r_2 \mathbf{F}_2^{\text{re}} = 0 \quad (4.17)$$

であり，角運動量保存則により，

$$\mathbf{x}_0 \times \mathbf{F}_0^{\text{shear}} + r_1 \mathbf{x}_1 \times \mathbf{F}_0^{\text{shear}} + r_2 \mathbf{x}_2 \times \mathbf{F}_0^{\text{shear}} = 0 \quad (4.18)$$

である．前節の座標系を対象とし， $\mathbf{H}_0 = \mathbf{x}_0 \times \mathbf{F}_0^{\text{shear}}, \mathbf{H}_1 = \mathbf{x}_1 \times \mathbf{F}_1^{\text{re}}, \mathbf{H}_2 = \mathbf{x}_2 \times \mathbf{F}_2^{\text{re}}$ とおくと，(4.17)(4.18) 式より，

$$\begin{aligned}r_1 &= \frac{-H_{0z} + H_{2z}}{H_{1z} - H_{2x}} = -\frac{1}{2}, \\ r_2 &= \frac{H_{0z} - H_{1z}}{H_{1z} - H_{2z}} = -\frac{1}{2}\end{aligned}$$

となる．よって，前節と同様に反作用が均等に割り振られていることが分かる．

4.2.3 三次元剪断モデルにおける剪断力

前述の剪断モデルを三次元に拡張することができる．まず，剪断力 F_i^{3D} を定義する．単純に四面体の各面の角度を剪断量とした場合，質点 P_i および P_l の相対位置ベクトルを $\mathbf{x}_1^{\text{rel}}, P_i$ および P_k の相対位置ベクトルを $\mathbf{x}_2^{\text{rel}}, P_i$ および P_j の相対位置ベ

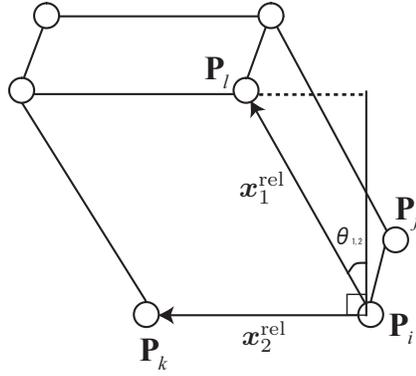


図 4.7: 三次元物体における剪断ひずみ

クトルを x_3^{rel} とし, $x_1^{\text{rel}}, x_2^{\text{rel}}, x_3^{\text{rel}}$ の単位ベクトルをそれぞれ $e_1^{\text{rel}}, e_2^{\text{rel}}, e_3^{\text{rel}}$ とし, x_1^{rel} と x_2^{rel} による三角形に対して二次元の場合と同様に定義した剪断量 $\tan \theta_{1,2}$, 同じく x_1^{rel} と x_3^{rel} での剪断量 $\tan \theta_{1,3}$, x_2^{rel} と x_3^{rel} での剪断量 $\tan \theta_{2,3}$ を,

$$\tan \theta_{1,2} = \frac{x_1^{\text{rel}} \cdot e_2^{\text{rel}}}{x_1^{\text{rel}} \cdot e_1^{\text{rel}}} = \frac{x_2^{\text{rel}} \cdot e_1^{\text{rel}}}{x_2^{\text{rel}} \cdot e_2^{\text{rel}}}, \quad (4.19)$$

$$\tan \theta_{1,3} = \frac{x_1^{\text{rel}} \cdot e_3^{\text{rel}}}{x_1^{\text{rel}} \cdot e_1^{\text{rel}}} = \frac{x_3^{\text{rel}} \cdot e_1^{\text{rel}}}{x_3^{\text{rel}} \cdot e_3^{\text{rel}}}, \quad (4.20)$$

$$\tan \theta_{2,3} = \frac{x_2^{\text{rel}} \cdot e_3^{\text{rel}}}{x_2^{\text{rel}} \cdot e_2^{\text{rel}}} = \frac{x_3^{\text{rel}} \cdot e_2^{\text{rel}}}{x_3^{\text{rel}} \cdot e_3^{\text{rel}}} \quad (4.21)$$

と定義する．例として, $\tan \theta_{1,2}$ を, 図 4.7 に示す．

ただし, この定義では, ある方向への剪断変形が他の方向への剪断変形量に影響を及ぼすという欠点がある．そこで, 次に, 他の方向の剪断変形の影響を受けない剪断量を以下に定義する．この定義では, 片方の稜線を通る面に射影し, 角度の計算を行う．すなわち, P_l 方向を正とする面 $P_i P_j P_k$ の法線ベクトルを e_A^{rel} , P_k 方向を正とする面 $P_i P_j P_l$ の法線ベクトルを e_B^{rel} , P_j 方向を正とする面 $P_i P_k P_l$ の法線ベクトルを e_C^{rel} とすると, 剪断量を,

$$\tan \theta_{1,2}^A = \frac{x_1^{\text{rel}} \cdot e_2^{\text{rel}}}{x_1^{\text{rel}} \cdot e_A^{\text{rel}}}, \quad (4.22)$$

$$\tan \theta_{1,3}^A = \frac{x_1^{\text{rel}} \cdot e_3^{\text{rel}}}{x_1^{\text{rel}} \cdot e_A^{\text{rel}}}, \quad (4.23)$$

$$\tan \theta_{2,3}^B = \frac{x_2^{\text{rel}} \cdot e_3^{\text{rel}}}{x_2^{\text{rel}} \cdot e_B^{\text{rel}}}, \quad (4.24)$$

$$\tan \theta_{2,1}^B = \frac{x_2^{\text{rel}} \cdot e_1^{\text{rel}}}{x_2^{\text{rel}} \cdot e_B^{\text{rel}}}, \quad (4.25)$$

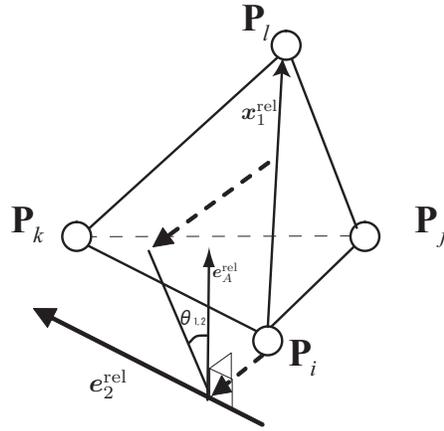


図 4.8: 三次元物体における各軸独立な剪断ひずみの定義

$$\tan \theta_{3,2}^C = \frac{\mathbf{x}_3^{\text{rel}} \cdot \mathbf{e}_2^{\text{rel}}}{\mathbf{x}_3^{\text{rel}} \cdot \mathbf{e}_C^{\text{rel}}}, \quad (4.26)$$

$$\tan \theta_{3,1}^C = \frac{\mathbf{x}_3^{\text{rel}} \cdot \mathbf{e}_1^{\text{rel}}}{\mathbf{x}_3^{\text{rel}} \cdot \mathbf{e}_C^{\text{rel}}} \quad (4.27)$$

で定義する．たとえば， $\tan \theta_{1,2}^A$ は，図 4.8 に表される量となる．

二次元の場合，質点と稜線を基点に角度を計算していた．この場合，稜線と面を基点に計算する形に拡張されている．定義した剪断量に対し，剪断力は，バネ・フォークトモデル・三要素モデル，一般化フォークトモデル等，任意の力学モデルを用いて計算することができる．各剪断量を $\gamma_{3D} = [\tan \theta_{1,2}^A \tan \theta_{1,3}^A \tan \theta_{2,3}^B \tan \theta_{2,1}^B \tan \theta_{3,2}^C \tan \theta_{3,1}^C]^T$ とし，力学モデルとしてフォークトモデルを選択した場合，剪断弾性率 G^{ela} ，剪断粘性率 G^{ela} ，剪断応力 $\lambda_i^{3D} = [\lambda_{A,1,2}^{3D} \lambda_{A,1,3}^{3D} \lambda_{B,2,3}^{3D} \lambda_{B,2,1}^{3D} \lambda_{C,3,2}^{3D} \lambda_{C,3,1}^{3D}]^T$ は，

$$\lambda_i^{3D} = -k \Delta \gamma_i^{3D} - c \dot{\gamma}_i^{3D} \quad (4.28)$$

となる．また質点 P_i にかかる剪断力 f_i^{3D} は，剪断応力と各剪断面の面積 S_A, S_B, S_C より，

$$f_i^{3D} = \frac{1}{4} \lambda_i^{3D} \cdot \begin{bmatrix} S_A \mathbf{e}_2^{\text{rel}} + S_B \mathbf{n}_A \\ S_A \mathbf{e}_3^{\text{rel}} + S_C \mathbf{n}_A \\ S_B \mathbf{e}_3^{\text{rel}} + S_C \mathbf{n}_B \\ S_B \mathbf{e}_1^{\text{rel}} + S_A \mathbf{n}_B \\ S_C \mathbf{e}_1^{\text{rel}} + S_A \mathbf{n}_C \\ S_C \mathbf{e}_2^{\text{rel}} + S_B \mathbf{n}_C \end{bmatrix} \quad (4.29)$$

とする．

4.3 体積モデル

4.3.1 二次元物体における体積変形

図 4.9 に示す三角形による二次元物体の体積変形について述べる．三角形 $P_i P_j P_k$ の符号付面積 $S_{i,j,k}$ は，

$$S_{i,j,k} = \frac{1}{2} \begin{vmatrix} P_{jx} - P_{ix} & P_{jy} - P_{iy} \\ P_{kx} - P_{ix} & P_{ky} - P_{iy} \end{vmatrix} \quad (4.30)$$

で与えられる． $S_{i,j,k}$ とその初期面積 $S_{i,j,k}^{\text{init}}$ に対し，体積ひずみ γ_{area} は，

$$\gamma_{\text{area}} = \frac{\Delta S_{i,j,k}}{S_{i,j,k}^{\text{init}}} \quad (4.31)$$

で表される．体積変形によって発生する内圧は，任意の力学モデルで計算できる，たとえば，フォークトモデルにおいて，体積弾性率 K^{ela} ，体積粘性率 K^{vis} とすると，内圧 P_{area} は，

$$P_{\text{area}} = -K^{\text{ela}} \gamma_{\text{area}} - K^{\text{vis}} \dot{\gamma}_{\text{area}} \quad (4.32)$$

となる．内圧は各稜線に対し外向きの法線方向に加わるため，稜線 $P_i P_j$ ， $P_i P_k$ ， $P_j P_k$ のそれぞれの長さを $l_{i,j}$ ， $l_{i,k}$ ， $l_{j,k}$ とし，各稜線の外向き法線ベクトルを $\mathbf{n}_{i,j}$ ， $\mathbf{n}_{i,k}$ ， $\mathbf{n}_{j,k}$ とすると，体積変形によって質点 P_i ， P_j ， P_k にかかる力 f_i^{area} ， f_j^{area} ， f_k^{area} は，

$$\begin{aligned} f_i^{\text{area}} &= \frac{1}{2} P_{\text{area}} (l_{i,k} \mathbf{n}_{i,k} + l_{i,j} \mathbf{n}_{i,j}), \\ f_j^{\text{area}} &= \frac{1}{2} P_{\text{area}} (l_{j,k} \mathbf{n}_{j,k} + l_{i,j} \mathbf{n}_{i,j}), \\ f_k^{\text{area}} &= \frac{1}{2} P_{\text{area}} (l_{i,k} \mathbf{n}_{i,k} + l_{j,k} \mathbf{n}_{j,k}) \end{aligned}$$

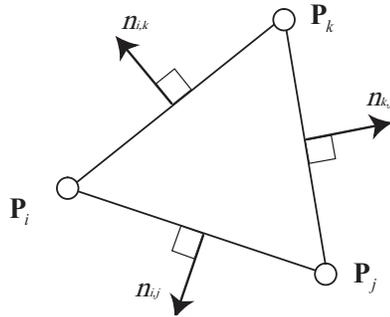


図 4.9: 二次元物体の体積変形

となる．

面積 $S_{i,j,k}$ は符号付であり，三角形が反転すると値がマイナスとなる．そのため，三角形が反転した状態で平衡状態とならず，初期形状に戻ろうとする力が働き，位相が保持される．これにより，大変形シミュレーションを安定におこなうことができる．

4.3.2 三次元物体における体積変形

二次元の場合と同様に，図 4.10 に示す四面体による三次元物体の体積変形について述べる．四面体 $P_i P_j P_k P_l$ の符号付体積 $V_{i,j,k,l}$ は，

$$V_{i,j,k,l} = \frac{1}{6} \begin{vmatrix} P_{jx} - P_{ix} & P_{jy} - P_{iy} & P_{jz} - P_{iz} \\ P_{kx} - P_{ix} & P_{ky} - P_{iy} & P_{kz} - P_{iz} \\ P_{lx} - P_{ix} & P_{ly} - P_{iy} & P_{lz} - P_{iz} \end{vmatrix} \quad (4.33)$$

で与えられる． $V_{i,j,k,l}$ とその初期体積 $V_{i,j,k,l}^{\text{init}}$ に対し，体積ひずみ γ_{vol} は，

$$\gamma_{\text{vol}} = \frac{\Delta V_{i,j,k,l}}{V_{i,j,k,l}^{\text{init}}} \quad (4.34)$$

で表される．これに対し発生する内圧は，任意の力学モデルで計算できる，たとえば，フォークトモデルにおいて，体積弾性率 K^{ela} ，体積粘性率 K^{vis} とすると，内圧 P_{vol} は，

$$P_{\text{vol}} = -K^{\text{ela}} \gamma_{\text{vol}} - K^{\text{vis}} \dot{\gamma}_{\text{vol}} \quad (4.35)$$

となる．

内圧は各面に対し外向きの法線方向に加わるため，三角面 $P_i P_j P_k$ の面積を $S_{i,j,k}$ とし，外向きの法線ベクトルを $\mathbf{n}_{i,j,k}$ とすると，体積変形によって質点 P_i ， P_j ， P_k ， P_l にかかる力 f_i^{vol} ， f_j^{vol} ， f_k^{vol} ， f_l^{vol} は，

$$\begin{aligned} f_i^{\text{vol}} &= \frac{1}{3} P_{\text{vol}} (S_{i,j,k} \mathbf{n}_{i,j,k} + S_{i,j,l} \mathbf{n}_{i,j,l} + S_{i,k,l} \mathbf{n}_{i,k,l}), \\ f_j^{\text{vol}} &= \frac{1}{3} P_{\text{vol}} (S_{i,j,k} \mathbf{n}_{i,j,k} + S_{i,j,l} \mathbf{n}_{i,j,l} + S_{j,k,l} \mathbf{n}_{j,k,l}), \\ f_k^{\text{vol}} &= \frac{1}{3} P_{\text{vol}} (S_{i,j,k} \mathbf{n}_{i,j,k} + S_{j,k,l} \mathbf{n}_{j,k,l} + S_{i,k,l} \mathbf{n}_{i,k,l}), \\ f_l^{\text{vol}} &= \frac{1}{3} P_{\text{vol}} (S_{j,k,l} \mathbf{n}_{j,k,l} + S_{i,j,l} \mathbf{n}_{i,j,l} + S_{i,k,l} \mathbf{n}_{i,k,l}) \end{aligned}$$

となる．二次元の場合と同様に， $V_{i,j,k,l}$ が符号付体積であるため，四面体が反転すると値がマイナスとなる．そのため，四面体が反転した状態で平衡状態とならず，初期

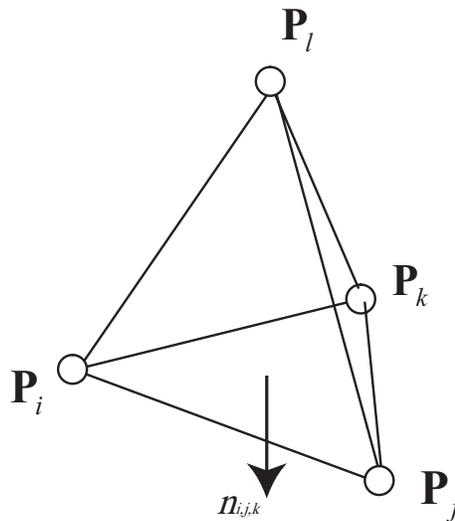


図 4.10: 三次元物体の体積変形

形状に戻ろうとする力が働き，位相が保持される．これにより，大変形シミュレーションを安定におこなうことができる．

4.4 CSMによる非圧縮性のモデリング

小麦粉生地は，変形の前後および変形中において体積が一定であるという特徴を持つ．これは非圧縮性と呼ばれる性質であり，体積弾性率が $K^{\text{ela}} = \infty$ かつ体積粘性率が $K^{\text{vis}} = \infty$ の時，物体は体積を一定に保ちながら変形する．しかし，前述の体積モデルにおいては，体積弾性率には有限の値しか設定できない．この問題の解決方法としては，体積および体積粘性率に有限の大きな値を設定するという手法が考えられる．しかし，その場合，パラメータの設定方法が明確ではなく，物体の変形特性に悪影響を与える恐れがある．そこで，本節において非圧縮性を表現する手法について述べる．

物体変形シミュレーションにおいて，物体の体積を一定に保つための手法として，CSM (Constraint Stabilization Method) [1] を用いた体積の拘束を提案する．CSM を用いることで，変形シミュレーション中に物体の体積が初期体積から外れても，臨界減衰的な挙動で初期体積と同じ体積に修正される．すなわち，振動的な挙動を生じることなく，物体の体積を拘束することができる．

4.4.1 面積一定制約

二次元物体の場合，三角面の面積を一定に保つことで非圧縮性を表現する．ここでは以下の図における三角形の物体に対して，CSMによる拘束を行う場合の数式を導出する．

図の物体において，三角形の面積を $S_{i,j,k}$ とし，三角形の自然状態での面積を $S_{i,j,k}^{\text{init}}$ とすると，拘束条件は $S_{i,j,k} - S_{i,j,k}^{\text{init}} = 0$ であり，すなわち

$$R = \frac{1}{2} \begin{vmatrix} P_{jx} - P_{ix} & P_{jy} - P_{iy} \\ P_{kx} - P_{ix} & P_{ky} - P_{iy} \end{vmatrix} - S_{i,j,k}^{\text{init}} = 0 \quad (4.36)$$

である．運動エネルギーを T とし，ラグランジュの未定乗数 λ をおくと，ラグランジアン L は，

$$L = T + \lambda R \quad (4.37)$$

であり，三つの質点の運動エネルギーと拘束条件によるラグランジアンは

$$L = \frac{1}{2}m_i(\dot{P}_{ix}^2 + \dot{P}_{iy}^2) + \frac{1}{2}m_j(\dot{P}_{jx}^2 + \dot{P}_{jy}^2) + \frac{1}{2}m_k(\dot{P}_{kx}^2 + \dot{P}_{ky}^2) + \lambda R \quad (4.38)$$

となる．ここで，ラグランジュの方程式から，

$$m_i\ddot{P}_{ix} - \frac{1}{2}(P_{jy} - P_{ky})\lambda = 0, \quad (4.39)$$

$$m_i\ddot{P}_{iy} - \frac{1}{2}(-P_{jx} + P_{kx})\lambda = 0, \quad (4.40)$$

$$m_j\ddot{P}_{jx} - \frac{1}{2}(P_{ky} - P_{iy})\lambda = 0, \quad (4.41)$$

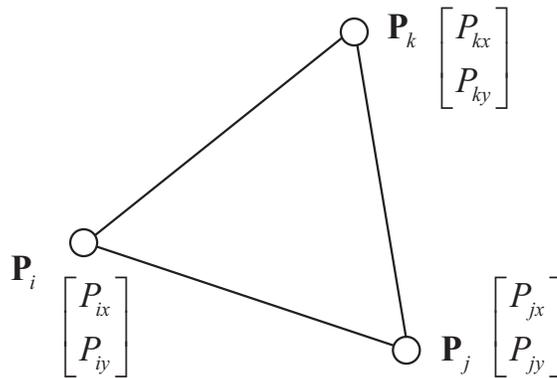


図 4.11: 単一の三角形による二次元物体

$$m_j \ddot{P}_{jy} - \frac{1}{2} (-P_{kx} + P_{ix}) \lambda = 0, \quad (4.42)$$

$$m_k \ddot{P}_{kx} - \frac{1}{2} (P_{iy} - P_{jy}) \lambda = 0, \quad (4.43)$$

$$m_k \ddot{P}_{ky} - \frac{1}{2} (-P_{ix} + P_{jx}) \lambda = 0 \quad (4.44)$$

となり，また，臨界減衰より，

$$\ddot{R} + 2\omega\dot{R} + \omega^2 R = 0 \quad (4.45)$$

とし， $\dot{P}_i = V_i$ ， $\dot{P}_j = V_j$ ， $\dot{P}_k = V_k$ とおくと，式 (4.39) ~ (4.45) より，一階の微分方程式を求めることができる．すなわち，

$$m_i \dot{V}_{ix} - \frac{1}{2} (P_{jy} - P_{ky}) \lambda = 0, \quad (4.46)$$

$$m_i \dot{V}_{iy} - \frac{1}{2} (-P_{jx} + P_{kx}) \lambda = 0, \quad (4.47)$$

$$m_j \dot{V}_{jx} - \frac{1}{2} (P_{ky} - P_{iy}) \lambda = 0, \quad (4.48)$$

$$m_j \dot{V}_{jy} - \frac{1}{2} (-P_{kx} + P_{ix}) \lambda = 0, \quad (4.49)$$

$$m_k \dot{V}_{kx} - \frac{1}{2} (P_{iy} - P_{jy}) \lambda = 0, \quad (4.50)$$

$$m_k \dot{V}_{ky} - \frac{1}{2} (-P_{ix} + P_{jx}) \lambda = 0, \quad (4.51)$$

$$\begin{aligned} & -\frac{1}{2} (P_{jy} - P_{ky}) \dot{V}_{ix} - \frac{1}{2} (-P_{jx} + P_{kx}) \dot{V}_{iy} \\ & -\frac{1}{2} (P_{ky} - P_{iy}) \dot{V}_{jx} - \frac{1}{2} (-P_{kx} + P_{ix}) \dot{V}_{jy} \\ & -\frac{1}{2} (P_{iy} - P_{jy}) \dot{V}_{kx} - \frac{1}{2} (-P_{ix} + P_{jx}) \dot{V}_{ky} \\ & = C(P_{ix}, \dots, P_{ky}, V_{ix}, \dots, V_{ky}) \end{aligned} \quad (4.52)$$

である．ただし，ここで，

$$\begin{aligned} & C(P_{ix}, P_{iy}, P_{jx}, P_{jy}, P_{kx}, P_{ky}, V_{ix}, V_{iy}, V_{jx}, V_{jy}, V_{kx}, V_{ky}) \\ & = 2\omega\dot{R} + \omega^2 R + \begin{vmatrix} V_{jx} - V_{ix} & V_{jy} - V_{iy} \\ V_{kx} - V_{ix} & V_{ky} - V_{iy} \end{vmatrix}, \\ & \dot{R}(P_{ix}, P_{iy}, P_{jx}, P_{jy}, P_{kx}, P_{ky}, V_{ix}, V_{iy}, V_{jx}, V_{jy}, V_{kx}, V_{ky}) \\ & = \frac{1}{2} \left(\begin{vmatrix} V_{jx} - V_{ix} & V_{jy} - V_{iy} \\ P_{kx} - P_{ix} & P_{ky} - P_{iy} \end{vmatrix} + \begin{vmatrix} P_{jx} - P_{ix} & P_{jy} - P_{iy} \\ V_{kx} - V_{ix} & V_{ky} - V_{iy} \end{vmatrix} \right) \end{aligned}$$

である．また，式 (4.46) ~ (4.52) を行列に直すと，

$$\begin{aligned}
 & \begin{bmatrix} I & & \\ & M & -A \\ & -A^T & \end{bmatrix} \begin{bmatrix} \dot{p}_N \\ \dot{v}_N \\ \lambda \end{bmatrix} \\
 = & \begin{bmatrix} & v_N & \\ & \mathbf{0} & \\ C(P_{ix}, \dots, P_{ky}, V_{ix}, \dots, V_{ky}) & & \end{bmatrix} \quad (4.53)
 \end{aligned}$$

となる．ただし，ここで

$$\begin{aligned}
 p_N &= [P_{ix} \ P_{iy} \ P_{jx} \ P_{jy} \ P_{kx} \ P_{ky}]^T, \\
 v_N &= [V_{ix} \ V_{iy} \ V_{jx} \ V_{jy} \ V_{kx} \ V_{ky}]^T, \\
 M &= \begin{bmatrix} m_i & 0 & 0 \\ 0 & m_j & 0 \\ 0 & 0 & m_k \end{bmatrix}, \\
 A &= \frac{1}{2} \begin{bmatrix} P_{jy} - P_{ky} \\ -P_{jx} + P_{kx} \\ P_{ky} - P_{iy} \\ -P_{kx} + P_{ix} \\ P_{iy} - P_{jy} \\ -P_{iy} + P_{jy} \end{bmatrix}
 \end{aligned}$$

である．(4.53) 式では左辺左側の行列の逆行列を求めることが可能であり， λ を求めることができる．CSM によって各節点に加わる力は λA であり，これを計算に適用することにより，物体の面積を振動を生じずに常にほぼ一定に保つことができると考えられる．

4.4.2 体積一定制約

前節と同様，三次元物体では，面積が体積に拡張される．すなわち図 4.12 に示す四面体 $P_i P_j P_k P_l$ の体積を一定に拘束する場合を考えると，拘束条件は，

$$R_V = \frac{1}{6} \begin{vmatrix} P_{jx} - P_{ix} & P_{jy} - P_{iy} & P_{jz} - P_{iz} \\ P_{kx} - P_{ix} & P_{ky} - P_{iy} & P_{kz} - P_{iz} \\ P_{lx} - P_{ix} & P_{ly} - P_{iy} & P_{lz} - P_{iz} \end{vmatrix} - V_{i,j,k,l}^{\text{init}} = 0 \quad (4.54)$$

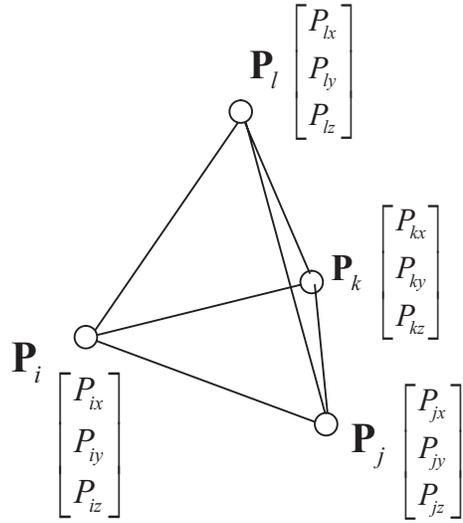


図 4.12: 単一の四面体による三次元物体

となる．四つの質点の運動エネルギーと拘束条件によるラグランジアンは

$$\begin{aligned}
 L_v &= \frac{1}{2}m_i (\dot{P}_{ix}^2 + \dot{P}_{iy}^2 + \dot{P}_{iz}^2) + \frac{1}{2}m_j (\dot{P}_{jx}^2 + \dot{P}_{jy}^2 + \dot{P}_{jz}^2) \\
 &+ \frac{1}{2}m_k (\dot{P}_{kx}^2 + \dot{P}_{ky}^2 + \dot{P}_{kz}^2) + \frac{1}{2}m_l (\dot{P}_{lx}^2 + \dot{P}_{ly}^2 + \dot{P}_{lz}^2) \\
 &+ \lambda R_V
 \end{aligned} \tag{4.55}$$

となる．また，臨界減衰より，

$$\ddot{R}_V + 2\omega \dot{R}_V + \omega^2 R_V = 0 \tag{4.56}$$

とし，ここで， $\dot{P}_i = V_i$ ， $\dot{P}_j = V_j$ ， $\dot{P}_k = V_k$ ， $\dot{P}_l = V_l$ とおくと，式(4.54)～(4.56)から，ラグランジュの方程式より，下記の一階の微分方程式が求まる．

$$\begin{aligned}
 m_j \dot{V}_{jx} - \frac{1}{6} \{ (P_{ky} - P_{iy})(P_{lz} - P_{iz}) - (P_{ly} - P_{iy})(P_{kz} - P_{iz}) \} \lambda &= 0, \\
 m_j \dot{V}_{jy} - \frac{1}{6} \{ (P_{kz} - P_{iz})(P_{lx} - P_{ix}) - (P_{kx} - P_{ix})(P_{lz} - P_{iz}) \} \lambda &= 0, \\
 m_j \dot{V}_{jz} - \frac{1}{6} \{ (P_{kx} - P_{ix})(P_{ly} - P_{iy}) - (P_{ky} - P_{iy})(P_{lx} - P_{ix}) \} \lambda &= 0, \\
 m_k \dot{V}_{kx} - \frac{1}{6} \{ (P_{jz} - P_{iz})(P_{ly} - P_{iy}) - (P_{jy} - P_{iy})(P_{lz} - P_{iz}) \} \lambda &= 0, \\
 m_k \dot{V}_{ky} - \frac{1}{6} \{ (P_{jx} - P_{ix})(P_{lz} - P_{iz}) - (P_{jz} - P_{iz})(P_{lx} - P_{ix}) \} \lambda &= 0,
 \end{aligned}$$

$$\begin{aligned}
m_k \dot{V}_{kz} - \frac{1}{6} \{ (P_{jy} - P_{iy})(P_{lx} - P_{ix}) - (P_{jx} - P_{ix})(P_{ly} - P_{iy}) \} \lambda &= 0, \\
m_l \dot{V}_{lx} - \frac{1}{6} \{ (P_{jy} - P_{iy})(P_{kz} - P_{iz}) - (P_{jz} - P_{iz})(P_{ky} - P_{iy}) \} \lambda &= 0, \\
m_l \dot{V}_{lx} - \frac{1}{6} \{ (P_{jz} - P_{iz})(P_{kx} - P_{ix}) - (P_{jx} - P_{ix})(P_{kz} - P_{iz}) \} \lambda &= 0, \\
m_l \dot{V}_{lz} - \frac{1}{6} \{ (P_{jx} - P_{ix})(P_{ky} - P_{iy}) - (P_{jy} - P_{iy})(P_{kx} - P_{ix}) \} \lambda &= 0, \\
m_i \dot{V}_{ix} - \frac{1}{6} \{ P_{jy}(P_{lz} - P_{kz}) + P_{ky}(P_{jz} - P_{lz}) + P_{ly}(P_{kz} - P_{jz}) \} \lambda &= 0, \\
m_i \dot{V}_{iy} - \frac{1}{6} \{ P_{jz}(P_{lx} - P_{kx}) + P_{kz}(P_{jx} - P_{lx}) + P_{lz}(P_{kx} - P_{jx}) \} \lambda &= 0, \\
m_i \dot{V}_{iy} - \frac{1}{6} \{ P_{jz}(P_{lx} - P_{kx}) + P_{kz}(P_{jx} - P_{lx}) + P_{lz}(P_{kx} - P_{jx}) \} \lambda &= 0, \\
m_i \dot{V}_{iz} - \frac{1}{6} \{ P_{jx}(P_{ly} - P_{ky}) + P_{kx}(P_{jy} - P_{ly}) + P_{lx}(P_{ky} - P_{jy}) \} \lambda &= 0.
\end{aligned}$$

さらに

$$\begin{aligned}
& -\frac{1}{6} \{ (P_{ky} - P_{iy})(P_{lz} - P_{iz}) - (P_{ly} - P_{iy})(P_{kz} - P_{iz}) \} \dot{V}_{jx} \\
& -\frac{1}{6} \{ (P_{kz} - P_{iz})(P_{lx} - P_{ix}) - (P_{kx} - P_{ix})(P_{lz} - P_{iz}) \} \dot{V}_{jy} \\
& -\frac{1}{6} \{ (P_{kx} - P_{ix})(P_{ly} - P_{iy}) - (P_{ky} - P_{iy})(P_{lx} - P_{ix}) \} \dot{V}_{jz} \\
& -\frac{1}{6} \{ (P_{jz} - P_{iz})(P_{ly} - P_{iy}) - (P_{jy} - P_{iy})(P_{lz} - P_{iz}) \} \dot{V}_{kx} \\
& -\frac{1}{6} \{ (P_{jx} - P_{ix})(P_{lz} - P_{iz}) - (P_{jz} - P_{iz})(P_{lx} - P_{ix}) \} \dot{V}_{ky} \\
& -\frac{1}{6} \{ (P_{jy} - P_{iy})(P_{lx} - P_{ix}) - (P_{jx} - P_{ix})(P_{ly} - P_{iy}) \} \dot{V}_{kz} \\
& -\frac{1}{6} \{ (P_{jy} - P_{iy})(P_{kz} - P_{iz}) - (P_{jz} - P_{iz})(P_{ky} - P_{iy}) \} \dot{V}_{lx} \\
& -\frac{1}{6} \{ (P_{jz} - P_{iz})(P_{kx} - P_{ix}) - (P_{jx} - P_{ix})(P_{kz} - P_{iz}) \} \dot{V}_{ly} \\
& -\frac{1}{6} \{ (P_{jx} - P_{ix})(P_{ky} - P_{iy}) - (P_{jy} - P_{iy})(P_{kx} - P_{ix}) \} \dot{V}_{lz} \\
& -\frac{1}{6} \{ P_{jy}(P_{lz} - P_{kz}) + P_{ky}(P_{jz} - P_{lz}) + P_{ly}(P_{kz} - P_{jz}) \} \dot{V}_{ix} \\
& -\frac{1}{6} \{ P_{jz}(P_{lx} - P_{kx}) + P_{kz}(P_{jx} - P_{lx}) + P_{lz}(P_{kx} - P_{jx}) \} \dot{V}_{iy} \\
& -\frac{1}{6} \{ P_{jx}(P_{ly} - P_{ky}) + P_{kx}(P_{jy} - P_{ly}) + P_{lx}(P_{ky} - P_{jy}) \} \dot{V}_{iz}
\end{aligned}$$

$$= C(\mathbf{P}_i, \mathbf{P}_j, \mathbf{P}_k, \mathbf{P}_l, \mathbf{V}_i, \mathbf{V}_j, \mathbf{V}_k, \mathbf{V}_l)$$

ここで,

$$\begin{aligned} & C(\mathbf{P}_i, \mathbf{P}_j, \mathbf{P}_k, \mathbf{P}_l, \mathbf{V}_i, \mathbf{V}_j, \mathbf{V}_k, \mathbf{V}_l) \\ &= \omega^2 R_V + 2\omega \dot{R}_V \\ &+ \frac{1}{6} \begin{vmatrix} V_{jx} - V_{ix} & V_{jy} - V_{iy} & V_{jz} - V_{iz} \\ V_{kx} - V_{ix} & V_{ky} - V_{iy} & V_{kz} - V_{iz} \\ P_{lx} - P_{ix} & P_{ly} - P_{iy} & P_{lz} - P_{iz} \end{vmatrix} \\ &+ \frac{1}{6} \begin{vmatrix} V_{jx} - V_{ix} & V_{jy} - V_{iy} & V_{jz} - V_{iz} \\ P_{kx} - P_{ix} & P_{ky} - P_{iy} & P_{kz} - P_{iz} \\ V_{lx} - V_{ix} & V_{ly} - V_{iy} & V_{lz} - V_{iz} \end{vmatrix} \\ &+ \frac{1}{6} \begin{vmatrix} P_{jx} - P_{ix} & P_{jy} - P_{iy} & P_{jz} - P_{iz} \\ V_{kx} - V_{ix} & V_{ky} - V_{iy} & V_{kz} - V_{iz} \\ V_{lx} - V_{ix} & V_{ly} - V_{iy} & V_{lz} - V_{iz} \end{vmatrix} \end{aligned}$$

また

$$\begin{aligned} \dot{R}_V &= \frac{1}{6} \begin{vmatrix} V_{jx} - V_{ix} & V_{jy} - V_{iy} & V_{jz} - V_{iz} \\ P_{kx} - P_{ix} & P_{ky} - P_{iy} & P_{kz} - P_{iz} \\ P_{lx} - P_{ix} & P_{ly} - P_{iy} & P_{lz} - P_{iz} \end{vmatrix} \\ &+ \frac{1}{6} \begin{vmatrix} P_{jx} - P_{ix} & P_{jy} - P_{iy} & P_{jz} - P_{iz} \\ V_{kx} - V_{ix} & V_{ky} - V_{iy} & V_{kz} - V_{iz} \\ P_{lx} - P_{ix} & P_{ly} - P_{iy} & P_{lz} - P_{iz} \end{vmatrix} \\ &+ \frac{1}{6} \begin{vmatrix} P_{jx} - P_{ix} & P_{jy} - P_{iy} & P_{jz} - P_{iz} \\ P_{kx} - P_{ix} & P_{ky} - P_{iy} & P_{kz} - P_{iz} \\ V_{lx} - V_{ix} & V_{ly} - V_{iy} & V_{lz} - V_{iz} \end{vmatrix} \end{aligned}$$

である．これらを行列の形に直すと，

$$\begin{bmatrix} I & & \\ & M & -A_v \\ & -A_v^T & \end{bmatrix} \begin{bmatrix} \dot{p}_N \\ \dot{v}_N \\ \lambda \end{bmatrix} = \begin{bmatrix} v_N \\ \mathbf{0} \\ C(\mathbf{P}_i, \mathbf{P}_j, \mathbf{P}_k, \mathbf{V}_i, \mathbf{V}_j, \mathbf{V}_k) \end{bmatrix} \quad (4.57)$$

となる．ただし，

$$M = \begin{bmatrix} m_i & 0 & 0 & 0 \\ 0 & m_j & 0 & 0 \\ 0 & 0 & m_k & 0 \\ 0 & 0 & 0 & m_l \end{bmatrix},$$

$$p_N = \begin{bmatrix} \mathbf{P}_i & \mathbf{P}_j & \mathbf{P}_k & \mathbf{P}_l \end{bmatrix}^T,$$

$$v_N = \begin{bmatrix} \mathbf{V}_i & \mathbf{V}_j & \mathbf{V}_k & \mathbf{V}_l \end{bmatrix}^T,$$

また

$$A_v = \frac{1}{6} \begin{bmatrix} P_{jy}(P_{lz} - P_{kz}) + P_{ky}(P_{jz} - P_{lz}) + P_{ly}(P_{kz} - P_{jz}) \\ P_{jz}(P_{lx} - P_{kx}) + P_{kz}(P_{jx} - P_{lx}) + P_{lz}(P_{kx} - P_{jx}) \\ P_{jx}(P_{ly} - P_{ky}) + P_{kx}(P_{jy} - P_{ly}) + P_{lx}(P_{ky} - P_{jy}) \\ (P_{ky} - P_{iy})(P_{lz} - P_{iz}) - (P_{ly} - P_{iy})(P_{kz} - P_{iz}) \\ (P_{kz} - P_{iz})(P_{lx} - P_{ix}) - (P_{kx} - P_{ix})(P_{lz} - P_{iz}) \\ (P_{kx} - P_{ix})(P_{ly} - P_{iy}) - (P_{ky} - P_{iy})(P_{lx} - P_{ix}) \\ (P_{jz} - P_{iz})(P_{ly} - P_{iy}) - (P_{jy} - P_{iy})(P_{lz} - P_{iz}) \\ (P_{jx} - P_{ix})(P_{lz} - P_{iz}) - (P_{jz} - P_{iz})(P_{lx} - P_{ix}) \\ (P_{jy} - P_{iy})(P_{lx} - P_{ix}) - (P_{jx} - P_{ix})(P_{ly} - P_{iy}) \\ (P_{jy} - P_{iy})(P_{kz} - P_{iz}) - (P_{jz} - P_{iz})(P_{ky} - P_{iy}) \\ (P_{jz} - P_{iz})(P_{kx} - P_{ix}) - (P_{jx} - P_{ix})(P_{kz} - P_{iz}) \\ (P_{jx} - P_{ix})(P_{ky} - P_{iy}) - (P_{jy} - P_{iy})(P_{kx} - P_{ix}) \end{bmatrix} \quad (4.58)$$

である．二次元の場合と同様，(4.57) 式の左辺左側の行列の逆行列を求めることが可能であり， λ を求めることができる．CSM によって各節点に加わる力は λA_v であり，これを計算に適用することにより，振動を生じずに物体の体積をほぼ一定に保つことができると考えられる．

4.4.3 パラメータの設定

CSM によって，体積弾性率 $K_n^{\text{ela}} = \infty$ および体積粘性率 $K_n^{\text{vis}} = \infty$ が表現できる．すなわち非圧縮性を持つ物体においては，CSM による体積拘束と，剪断モデルを用いてシミュレーションを行えばよい．体積弾性率および体積粘性率が決定しているため，剪断モデルのパラメータは，剪断弾性率および剪断粘性率を設定するか，もしくは，伸張弾性率および伸張粘性率から求めることができる．著者らのプロジェクトでは，小麦粉生地伸張弾性率および伸張粘性率の抽出を行っている．そこで，これを利用できるように，剪断弾性率に変換する．

力学的等方性を持つ物体において，

$$E_n^{\text{ela}} = 2G_n^{\text{ela}} (1 + \mu_n^{\text{ela}}) = 3K_n^{\text{ela}} (1 - 2\mu_n^{\text{ela}}), \quad (4.59)$$

$$E_n^{\text{vis}} = 2G_n^{\text{vis}} (1 + \mu_n^{\text{vis}}) = 3K_n^{\text{vis}} (1 - 2\mu_n^{\text{vis}}) \quad (4.60)$$

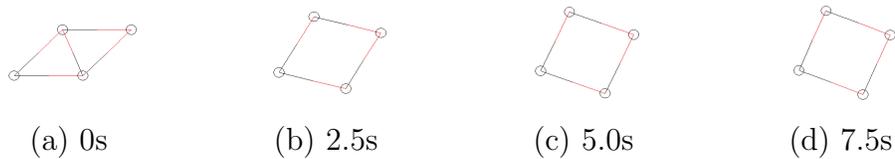


図 4.13: 剪断フォークトモデル



図 4.14: 剪断三要素モデル

が成り立つ．これに体積弾性率 $G_n^{\text{ela}} = \infty$, $G_n^{\text{vis}} = \infty$ を代入することで，

$$G_n^{\text{ela}} = \frac{1}{3} E_n^{\text{ela}}, \quad (4.61)$$

$$G_n^{\text{vis}} = \frac{1}{3} E_n^{\text{vis}} \quad (4.62)$$

となる．

4.5 シミュレーション

4.5.1 四角形剪断モデルによる物体変形シミュレーション

四角形剪断モデルによる物体変形シミュレーションを行った．図 4.13 および図 4.14 がそれぞれのシミュレーション例である．フォークトモデルでは残留変位を生じず，振動が減衰しながら初期形状へと変形し，三要素モデルでは，戻り変位と残留変位の両方が生じており，それぞれの特徴が剪断方向に現れていることが確認できる．

4.5.2 三角形剪断モデルおよび体積モデルによる弾性物体，粘塑性物体，レオロジー物体の変形シミュレーション

四角形剪断モデルおよび体積モデルの組み合わせでは平衡状態が一意に決まらないという問題がある．そのため，弾性物体のシミュレーションにおいて，物体が初期形状に復元しない可能性がある．例えば，正方形形状の物体が，面積の同じ長方形

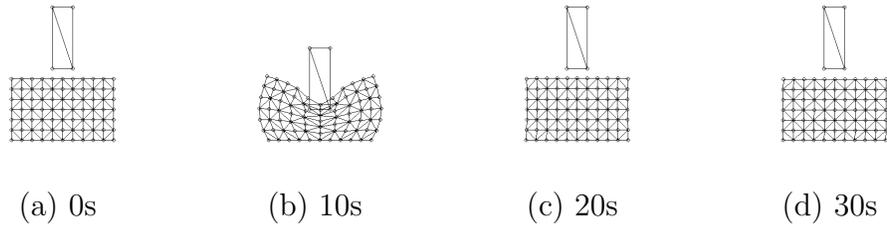


図 4.15: 三角形剪断モデルおよび体積モデルによる弾性物体

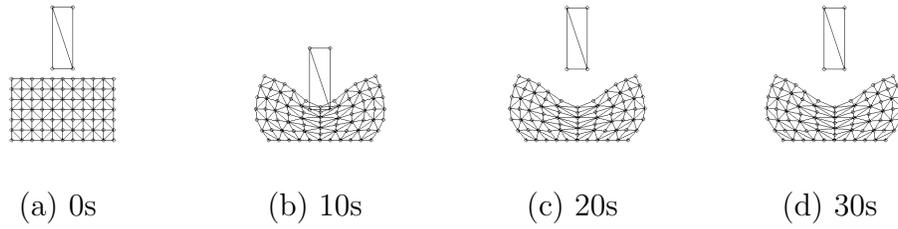


図 4.16: 三角形剪断モデルおよび体積モデルによる粘塑性物体

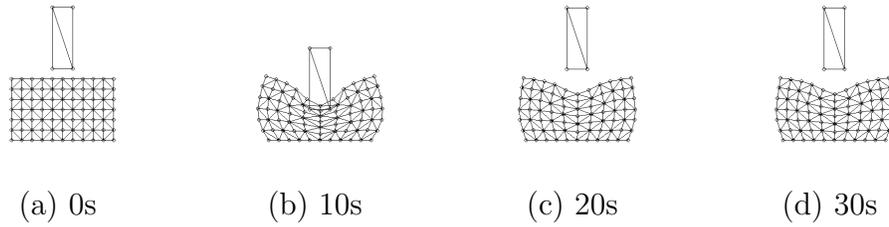


図 4.17: 三角形剪断モデルおよび体積モデルによるレオロジー物体

に変形した場合，剪断モデルおよび体積モデルの双方で発生する力が0となり，平衡状態となる．ただし，三角形に適用した剪断モデルの場合，この問題は発生しない．三角形剪断モデルと体積モデルを組み合わせることで，弾性物体における平衡形状を一意に決定することが可能である．

三角形剪断モデルおよび体積モデルによる物体変形シミュレーションを行った．弾性物体，塑性物体，レオロジー物体のシミュレーション例をそれぞれ図 4.15，図 4.16，図 4.17 に示す．シミュレーション時間は 30s であり，0s から 10s までの時間，上部の剛体を押し下げ，10s から 20s をかけて，上部の剛体を初期位置に戻した．体積モデルにおける圧力はフォークトモデルで表現し，パラメータはいずれも $K_1^{\text{ela}} = 10$ ， $K_1^{\text{vis}} = 1$ とした．剪断モデルにおける剪断応力は三要素モデルで表現し，パラメータは弾性物体においては $G_1^{\text{ela}} = 1$ ， $G_1^{\text{vis}} = 1$ ， $G_2^{\text{vis}} = 100000$ ，粘塑性物体において

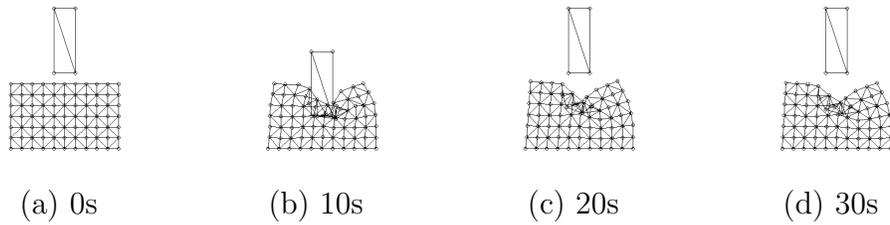


図 4.18: パーティクルベースモデルによる弾性物体のシミュレーション

は $G_1^{\text{ela}} = 0$, $G_1^{\text{vis}} = 5$, $G_2^{\text{vis}} = 5$, レオロジー物体においては $G_1^{\text{ela}} = 1$, $G_1^{\text{vis}} = 1$, $G_2^{\text{vis}} = 10$ とした. また, 物体の質量は 30 とし, 質点間の間隔は 5 とした.

通常のパーティクルベースモデルでは, このような局所的な大変形において, 図 4.18 に示すように三角形が反転し, 元の形状に戻らなくなることがある. 三角形剪断モデルおよび体積モデルでは大変形においても安定にシミュレーションが行われている.

4.6 結言

本章では, 大変形シミュレーションが可能な変形モデルとして, 剪断と非圧縮性を考慮した小麦粉生地の変形モデルを提案した. 剪断変形と体積変形を考慮することで, 連続体モデルのパラメータが適用可能なパーティクルベースモデルが構築可能である. さらに, 体積変形が符号付きの量であるため, 大変形に対応し, 安定にリアルタイム変形シミュレーションが可能なレオロジー物体変形モデルを構築できる.

レオロジー変形の動的モデリング

Dynamic Modeling of Rheological Deformation

平井 慎一, 友國 誠至 (立命館大学)

Shinichi HIRAI and Seiji TOMOKUNI

Ritsumeikan Univ., Kusatsu, Shiga 525-8577, Japan

We describe continua modeling of a rheologically deformable object. Rheological 2D deformation is formulated based on continua modeling. We show a simple simulation to demonstrate the modeling capability.

Key Words: rheology, modeling, deformation, continua, dynamic

1. Introduction

Most food and biological tissue show rheological nature in their deformation. Modeling and identification of these rheologically deformable objects are needed in virtual reality, especially, surgery simulation and mastication simulation. We have applied a particle-based approach to the modeling of rheological objects⁽¹⁾. Unfortunately, physical meaning of model parameters is unclear in the particle-based approach, resulting the difficulty in identification of model parameters. Note that continua modeling stands on a clear foundation. In this paper, we apply the continua modeling to 2D rheological deformation to build a dynamic model of a rheological object.

2. Rheological objects

Objects deform in response to forces applied to the objects. Objects can be categorized into three groups with respect to their deformation. Assume that a natural shape of an object is as given in Figure 1-(a). On applying external forces, the object deforms as in Figure 1-(b). Let us release the applied force and examine the stable shape after the release. Deformation of *viscoelastic objects* is completely lost and their stable shape coincides with their natural shape, as illustrated in Figure 1-(c). Namely, viscoelastic objects have no *residual deformation*. Deformation of *plastic objects* completely remains and their stable shape coincides with their deformed shape under the applied forces, as shown in Figure 1-(d). Namely, plastic objects have no *bouncing deformation*. Objects with residual deformation and bouncing deformation are referred to as *rheological objects*. Deformation of rheological objects is partially lost after the applied forces are released, as illustrated in Figure 1-(e). Various objects including foods and tissues are categorized into rheological objects.

3. Dynamic modeling of 2D rheological object

Let $\boldsymbol{\sigma}$ be a pseudo stress vector and $\boldsymbol{\varepsilon}$ be a pseudo strain vector. Stress-strain relationship of 2D rheological deformation is formulated as follows:

$$\boldsymbol{\sigma}(t) = \int_0^t R(t-t') \dot{\boldsymbol{\varepsilon}}(t') dt', \quad (1)$$

where 3×3 matrix $R(t-t')$ is referred to as a *relaxation matrix*, which determines the nature of a 2D rheological deformation. The relaxation matrix of 2D isotropic rheological deformation is formulated as

$$R(t-t') = r_\lambda(t-t')I_\lambda + r_\mu(t-t')I_\mu \quad (2)$$

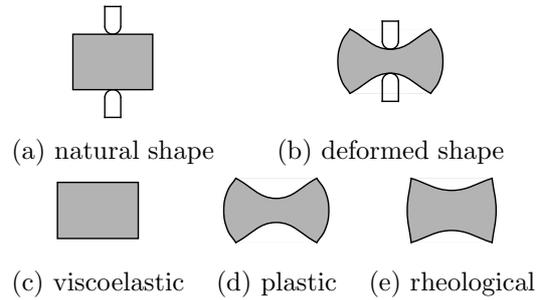


Fig.1 Viscoelastic object, plastic object, and rheological object

where

$$r_\lambda(t-t') = \lambda_{\text{ela}} \exp \left\{ -\frac{\lambda_{\text{ela}}}{\lambda_{\text{vis}}} (t-t') \right\},$$

$$r_\mu(t-t') = \mu_{\text{ela}} \exp \left\{ -\frac{\mu_{\text{ela}}}{\mu_{\text{vis}}} (t-t') \right\}.$$

Elasticity of the object is specified by two elastic moduli λ^{ela} and μ^{ela} while its viscosity is specified by two viscous moduli λ^{vis} and μ^{vis} . Matrices I_λ and I_μ are matrix representations of isotropic tensors, which are given as follows in 2D deformation:

$$I_\lambda = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad I_\mu = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

The stress-strain relationship can be converted into a relationship between a set of forces applied to nodal points and a set of displacements of the points. Let \mathbf{u}_N be a set of displacements of nodal points. Let J_λ and J_μ are connection matrices, which can be geometrically determined by object coordinate components of nodal points. Replacing I_λ by J_λ , I_μ by J_μ , and $\boldsymbol{\varepsilon}$ by \mathbf{u}_N in the stress-strain relationship (1) of a rheological object yields a set of rheological forces applied to nodal points as follows:

$$\text{rheological force} = J_\lambda \mathbf{w}_\lambda + J_\mu \mathbf{w}_\mu \quad (3)$$

where

$$\mathbf{w}_\lambda = \int_0^t \lambda_{\text{ela}} \exp \left\{ -\frac{\lambda_{\text{ela}}}{\lambda_{\text{vis}}} (t-t') \right\} \dot{\mathbf{u}}_N(t') dt',$$

Chapter 5

Continuous Modeling of Inelastic Deformation

5.1 Viscoplastic Deformation

5.1.1 Maxwell model

Let us formulate viscoplastic deformation of an object. Simple viscoplastic deformation can be described by Maxwell model illustrated in Figure 5.1. Maxwell model consists of an elastic element and a viscous element connected in serial. Let E be Young's modulus, which represents the elastic element, and c be viscous modulus, which characterizes the viscous element. Let ε^{ela} and ε^{vis} be strains at the elastic and viscous elements. Let ε be strain of the Maxwell model and σ be stress applied to the model. Strain ε coincides to the sum of strains at the two elements. Stress σ is equal to the stress caused by the elastic element as well as the stress caused by the viscous element. That is,

$$\begin{aligned}\varepsilon &= \varepsilon^{\text{ela}} + \varepsilon^{\text{vis}}, \\ \sigma &= E\varepsilon^{\text{ela}}, \quad \sigma = c\dot{\varepsilon}^{\text{vis}}.\end{aligned}$$

From the above equations, we have the following first order differential equation:

$$\dot{\sigma} + \frac{E}{c}\sigma = E\dot{\varepsilon}.$$



Figure 5.1: Maxwell model of viscoplastic deformation

Solving the above differential equation, stress at time t is described as follows in a convolution form:

$$\sigma(t) = \int_0^t E e^{-\frac{E}{c}(t-t')} \dot{\varepsilon}(t') dt'. \quad (5.1)$$

In general,

$$\sigma(t) = \int_0^t r(t-t') \dot{\varepsilon}(t') dt'. \quad (5.2)$$

Function $r(t-t')$ is referred to as a *relaxation function*.

Let us reformulate eq.(5.1) using Laplace transform. Let $\sigma(s)$, $\varepsilon(s)$, $\varepsilon^{\text{ela}}(s)$, and $\varepsilon^{\text{vis}}(s)$ are Laplace transforms of $\sigma(t)$, $\varepsilon(t)$, $\varepsilon^{\text{ela}}(t)$, and $\varepsilon^{\text{vis}}(t)$. Then, we have

$$\begin{aligned} \varepsilon(s) &= \varepsilon^{\text{ela}}(s) + \varepsilon^{\text{vis}}(s), \\ \sigma(s) &= E\varepsilon^{\text{ela}}(s), \quad \sigma(s) = cs\varepsilon^{\text{vis}}(s). \end{aligned}$$

From the above equations, we have

$$\sigma(s) = \frac{E}{s + E/c} s\varepsilon(s).$$

Applying the inverse Laplace transform to the above equation successfully yields eq.(5.1).

5.1.2 Deformation of 1D viscoplastic beam

Let us apply finite element approach to the deformation of 1D viscoplastic beam. The initial length of the beam is given by L . Assume that its cross-sectional area A is constant. The left end point of the beam is fixed to space while an external force $f(t)$ is applied to its right end point. Divide region $[0, L]$ into 4 regions with constant intervals specified by h . A set of displacements at five nodal points is given in a vector form by $\mathbf{u}_N = [u_0, u_1, u_2, u_3, u_4]^T$.

Recall that the elastic deformation is characterized by Young's modulus E , which defines a stress-strain relationship: $\sigma = E\varepsilon$. Finite element approximation yields a set of elastic forces applied to nodal points, which is described as $K\mathbf{u}_N$ in Section 3.2.2. Note that a set of elastic forces applied to nodal points is reformulated as

$$\text{elastic force} = E J \mathbf{u}_N$$

where

$$J = \frac{A}{h} \begin{bmatrix} 1 & -1 & & & & \\ -1 & 2 & -1 & & & \\ & -1 & 2 & -1 & & \\ & & -1 & 2 & -1 & \\ & & & -1 & 2 & -1 \\ & & & & -1 & 1 \end{bmatrix}.$$

Consequently, replacing strain ε in the stress-strain relationship by $J\mathbf{u}_N$ yields a set of elastic forces applied to nodal points. Note that matrix J can be determined by geometric quantities alone. Matrix J is time-invariant and includes no physical parameters.

Recall that a stress-strain relationship in viscoplastic deformation is described by eq.(5.1). Thus, replacing strain ε in the equation by $J\mathbf{u}_N$, we can derive a set of viscoplastic forces applied to nodal points as follows:

$$\text{viscoplastic force} = \int_0^t E e^{-\frac{E}{c}(t-t')} J \dot{\mathbf{u}}_N(t') dt'.$$

Let us introduce the following vector:

$$\mathbf{w} = \int_0^t E e^{-\frac{E}{c}(t-t')} \dot{\mathbf{u}}_N(t') dt'.$$

A set of viscoplastic forces applied to nodal points is then simply described as

$$\text{viscoplastic force} = J \mathbf{w}.$$

Introduce the velocity vector $\mathbf{v}_N = \dot{\mathbf{u}}_N$ and replacing elastic force $K\mathbf{u}_N$ in eq.(3.18) by the viscoplastic force given in the above equation, a set of dynamic equations of nodal points is described as follows:

$$-J\mathbf{w} + \lambda \mathbf{a} + \mathbf{f} - M\dot{\mathbf{v}}_N = \mathbf{0}. \quad (5.3)$$

where $\mathbf{a} = [1, 0, 0, 0, 0]^T$ and $\mathbf{f} = [0, 0, 0, 0, f(t)]$. Lagrange multiplier λ denotes the magnitude of a constraint force corresponding to a geometric constraint $\mathbf{a}^T \mathbf{u}_N = 0$. Differentiating \mathbf{w} with respect to time t , we have the following equation:

$$\dot{\mathbf{w}} = -\frac{E}{c} \mathbf{w} + E \dot{\mathbf{u}}_N = -\frac{E}{c} \mathbf{w} + E \mathbf{v}_N.$$

Applying the CSM to a geometric constraint $\mathbf{a}^T \mathbf{u}_N = 0$ yields the following differential equation:

$$\mathbf{a}^T \ddot{\mathbf{u}}_N + \mathbf{a}^T (2\omega \dot{\mathbf{u}}_N + \omega^2 \mathbf{u}_N) = 0.$$

Consequently, the equations of motion and the above differential equations can be described as follows:

$$\begin{aligned} \dot{\mathbf{u}}_N &= \mathbf{v}_N, \\ M \dot{\mathbf{v}}_N - \mathbf{a} \lambda &= -J \mathbf{w} + \mathbf{f}, \\ -\mathbf{a}^T \dot{\mathbf{v}}_N &= \mathbf{a}^T (2\omega \mathbf{v}_N + \omega^2 \mathbf{u}_N), \\ \dot{\mathbf{w}} &= -\frac{E}{c} \mathbf{w} + E \mathbf{v}_N. \end{aligned} \tag{5.4}$$

Namely,

$$\begin{bmatrix} I & & & & \\ & M & -\mathbf{a} & & \\ & -\mathbf{a}^T & & & \\ & & & I & \\ & & & & I \end{bmatrix} \begin{bmatrix} \dot{\mathbf{u}}_N \\ \dot{\mathbf{v}}_N \\ \lambda \\ \dot{\mathbf{w}} \end{bmatrix} = \begin{bmatrix} \mathbf{v}_N \\ -J \mathbf{w} + \mathbf{f} \\ \mathbf{a}^T (2\omega \mathbf{v}_N + \omega^2 \mathbf{u}_N) \\ -\frac{E}{c} \mathbf{w} + E \mathbf{v}_N \end{bmatrix}. \tag{5.5}$$

Note that the above linear equation is solvable since the matrix is regular, implying that we can compute $\dot{\mathbf{u}}_N$, $\dot{\mathbf{v}}_N$, and $\dot{\mathbf{w}}$. As a result, we can sketch \mathbf{u}_N , \mathbf{v}_N , and \mathbf{w} using the Euler method or the Runge-Kutta method. In addition, as the matrix in the above equation is symmetric and block diagonal, we can compute its inverse matrix numerically in an efficient manner.

5.2 Rheological Deformation

5.2.1 Three-element model

Let us formulate rheological deformation of an object. Simple rheological deformation can be described by three-element model illustrated in Figure

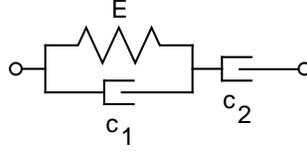


Figure 5.2: Three-element model of rheological deformation

5.2. Three-element model consists of a Voigt element and a viscous element connected in serial. Let E and c_1 be Young's modulus of an elastic element and viscous modulus in the Voigt element. Let c_2 be viscous modulus of the viscous element. Let $\varepsilon^{\text{voigt}}$ and ε^{vis} be strains at the Voigt and viscous elements. Let ε be strain of the three-element model and σ be stress applied to the model. Strain ε coincides to the sum of strains at the two elements. Stress σ is equal to the stress caused by the Voigt element as well as the stress caused by the viscous element. That is,

$$\begin{aligned}\varepsilon &= \varepsilon^{\text{voigt}} + \varepsilon^{\text{vis}}, \\ \sigma &= E\varepsilon^{\text{voigt}} + c_1\dot{\varepsilon}^{\text{voigt}}, \quad \sigma = c_2\dot{\varepsilon}^{\text{vis}}.\end{aligned}$$

From the above equations, we have the following first order differential equation on stress σ :

$$\dot{\sigma} + \frac{E}{c_1 + c_2}\sigma = \frac{c_1 c_2}{c_1 + c_2}\ddot{\varepsilon} + \frac{E c_2}{c_1 + c_2}\dot{\varepsilon}.$$

Solving the above differential equation, stress at time t is described as follows in a convolution form:

$$\sigma(t) = \int_0^t r(t-t') \dot{\varepsilon}(t') dt' \quad (5.6)$$

where

$$r(t-t') = \frac{E c_2}{c_1 + c_2} e^{-\frac{E}{c_1 + c_2}(t-t')} \left(1 + \frac{c_1}{E} \frac{d}{dt} \right). \quad (5.7)$$

Let us reformulate eqs.(5.6) and (5.7) using Laplace transform. Let $\sigma(s)$, $\varepsilon(s)$, $\varepsilon^{\text{voigt}}(s)$, and $\varepsilon^{\text{vis}}(s)$ are Laplace transforms of $\sigma(t)$, $\varepsilon(t)$, $\varepsilon^{\text{voigt}}(t)$, and $\varepsilon^{\text{vis}}(t)$. Then, we have

$$\begin{aligned}\varepsilon(s) &= \varepsilon^{\text{voigt}}(s) + \varepsilon^{\text{vis}}(s), \\ \sigma(s) &= E\varepsilon^{\text{voigt}}(s) + c_1 s \varepsilon^{\text{voigt}}(s), \quad \sigma(s) = c_2 s \varepsilon^{\text{vis}}(s).\end{aligned}$$

From the above equations, we have

$$\sigma(s) = \left(\frac{c_2}{c_1 + c_2} \right) \frac{E}{s + E/(c_1 + c_2)} \left(1 + \frac{c_1}{E}s \right) s\varepsilon(s).$$

Applying the inverse Laplace transform to the above equation successfully yields eqs.(5.6) and (5.7).

5.2.2 Deformation of 1D rheological beam

Let us apply finite element approach to the deformation of 1D rheological beam. The initial length of the beam is given by L . Assume that its cross-sectional area A is constant. The left end point of the beam is fixed to space while an external force $f(t)$ is applied to its right end point. Divide region $[0, L]$ into 4 regions with constant intervals specified by h . A set of displacements at five nodal points is given in a vector form by $\mathbf{u}_N = [u_0, u_1, u_2, u_3, u_4]^T$.

Recall that a stress-strain relationship in rheological deformation is described by eq.(5.6). Thus, replacing strain ε in the equation by $J\mathbf{u}_N$, we can derive a set of rheological forces applied to nodal points as follows:

$$\begin{aligned} \text{rheological force} &= \int_0^t r(t-t') J\dot{\mathbf{u}}_N(t') dt' \\ &= \int_0^t \frac{Ec_2}{c_1 + c_2} e^{-\frac{E}{c_1+c_2}(t-t')} \left(1 + \frac{c_1}{E} \frac{d}{dt} \right) J\dot{\mathbf{u}}_N(t') dt'. \end{aligned}$$

Let us introduce the following vector:

$$\mathbf{w} = \int_0^t \frac{Ec_2}{c_1 + c_2} e^{-\frac{E}{c_1+c_2}(t-t')} \left(\dot{\mathbf{u}}_N(t') + \frac{c_1}{E} \ddot{\mathbf{u}}_N(t') \right) dt' - \frac{c_1 c_2}{c_1 + c_2} \dot{\mathbf{u}}_N.$$

A set of rheological forces applied to nodal points is then described as

$$\text{rheological force} = J \left(\mathbf{w} + \frac{c_1 c_2}{c_1 + c_2} \dot{\mathbf{u}}_N \right).$$

Introducing the velocity vector $\mathbf{v}_N = \dot{\mathbf{u}}_N$ and replacing elastic force $K\mathbf{u}_N$ in eq.(3.18) by the rheological force given in the above equation, a set of dynamic equations of nodal points is described as follows:

$$-J \left(\mathbf{w} + \frac{c_1 c_2}{c_1 + c_2} \dot{\mathbf{u}}_N \right) + \lambda \mathbf{a} + \mathbf{f} - M\dot{\mathbf{v}}_N = \mathbf{0}. \quad (5.8)$$

Differentiating \mathbf{w} with respect to time t , we have the following equation:

$$\dot{\mathbf{w}} = -\frac{E}{c_1 + c_2}\mathbf{w} + \frac{Ec_2}{c_1 + c_2}\mathbf{v}_N.$$

Consequently, the equations of motion and the above differential equations can be described as follows:

$$\begin{aligned} \dot{\mathbf{u}}_N &= \mathbf{v}_N, \\ M\dot{\mathbf{v}}_N - \mathbf{a}\lambda &= -J\left(\mathbf{w} + \frac{c_1c_2}{c_1 + c_2}\mathbf{v}_N\right) + \mathbf{f}, \\ -\mathbf{a}^T\dot{\mathbf{v}}_N &= \mathbf{a}^T(2\omega\mathbf{v}_N + \omega^2\mathbf{u}_N), \\ \dot{\mathbf{w}} &= -\frac{E}{c_1 + c_2}(\mathbf{w} - c_2\mathbf{v}_N). \end{aligned} \quad (5.9)$$

Namely,

$$\begin{bmatrix} I & & & \\ & M & -\mathbf{a} & \\ & -\mathbf{a}^T & & \\ & & & I \end{bmatrix} \begin{bmatrix} \dot{\mathbf{u}}_N \\ \dot{\mathbf{v}}_N \\ \lambda \\ \dot{\mathbf{w}} \end{bmatrix} = \begin{bmatrix} \mathbf{v}_N \\ -J\left(\mathbf{w} + \frac{c_1c_2}{c_1 + c_2}\mathbf{v}_N\right) + \mathbf{f} \\ \mathbf{a}^T(2\omega\mathbf{v}_N + \omega^2\mathbf{u}_N) \\ -\frac{E}{c_1 + c_2}(\mathbf{w} - c_2\mathbf{v}_N) \end{bmatrix}. \quad (5.10)$$

Note that the above linear equation is solvable since the matrix is regular, implying that we can compute $\dot{\mathbf{u}}_N$, $\dot{\mathbf{v}}_N$, and $\dot{\mathbf{w}}$. As a result, we can sketch \mathbf{u}_N , \mathbf{v}_N , and \mathbf{w} using the Euler method or the Runge-Kutta method. In addition, as the matrix in the above equation is symmetric and block diagonal, we can compute its inverse matrix numerically in an efficient manner.

5.3 Multi-dimensional Inelastic Deformation

5.3.1 Deformation models

Recall that the stress-strain relationship of an elastic object can be specified by a constant E . In addition, 2D isotropic elastic deformation can be formulated as follows:

$$\boldsymbol{\sigma} = (\lambda I_\lambda + \mu I_\mu)\boldsymbol{\varepsilon} \quad (5.11)$$

where λ and μ denote Lamé's constants and

$$I_\lambda = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad I_\mu = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (5.12)$$

Matrices I_λ and I_μ originate from the isotropy of the object deformation. Elastic nature of a deformable object can be specified by two constants: λ and μ . These constants determine normal elasticity and shear elasticity.

Recall that the stress-strain relationship of a viscoelastic object can be specified by an operator: $E + c d/dt$. From the above observation, replacing two elastic constants in eq.(5.11) by two viscoelastic operators yields 2D isotropic viscoelastic deformation as follows:

$$\boldsymbol{\sigma} = (\lambda I_\lambda + \mu I_\mu) \boldsymbol{\varepsilon} \quad (5.13)$$

where

$$\lambda = \lambda^{\text{ela}} + \lambda^{\text{vis}} \frac{d}{dt}, \quad \mu = \mu^{\text{ela}} + \mu^{\text{vis}} \frac{d}{dt}.$$

Two constants λ^{ela} and μ^{ela} specify elasticity of the object while λ^{vis} and μ^{vis} describe its viscosity. Operator $\lambda^{\text{ela}} + \lambda^{\text{vis}} d/dt$ characterizes normal viscoelasticity of the object while its shear viscoelasticity is described in $\mu^{\text{ela}} + \mu^{\text{vis}} d/dt$.

Recall that the stress-strain relationship of a viscoplastic object can be specified by a relaxation function $r(t-t')$ and eq.(5.2). Then, 2D viscoplastic deformation can be described as follows:

$$\boldsymbol{\sigma}(t) = \int_0^t R(t-t') \dot{\boldsymbol{\varepsilon}}(t') dt', \quad (5.14)$$

where 3×3 matrix $R(t-t')$ is referred to as a *relaxation matrix*, which determines the nature of a 2D viscoplastic deformation. Replacing two elastic constants in eq.(5.11) by two relaxation functions yields a relaxation matrix of 2D isotropic viscoplastic deformation:

$$R(t-t') = r_\lambda(t-t') I_\lambda + r_\mu(t-t') I_\mu \quad (5.15)$$

where

$$\begin{aligned} r_\lambda(t-t') &= \lambda^{\text{ela}} \exp \left\{ -\frac{\lambda^{\text{ela}}}{\lambda^{\text{vis}}} (t-t') \right\}, \\ r_\mu(t-t') &= \mu^{\text{ela}} \exp \left\{ -\frac{\mu^{\text{ela}}}{\mu^{\text{vis}}} (t-t') \right\}. \end{aligned}$$

Two constants λ^{ela} and μ^{ela} specify elasticity of the object while λ^{vis} and μ^{vis} describe its plasticity. Relaxation function $r_\lambda(t-t')$ characterizes normal

viscoplasticity of the object while its shear viscoplasticity is described in $r_\mu(t - t')$.

Recall that the stress-strain relationship of a rheological object can be specified by a relaxation function $r(t' - t)$ and eq.(5.6). Replacing two elastic constants in eq.(5.11) by two relaxation functions yields a relaxation matrix of 2D isotropic rheological deformation:

$$R(t - t') = r_\lambda^{\text{rheo}}(t - t')I_\lambda + r_\mu^{\text{rheo}}(t - t')I_\mu \quad (5.16)$$

where

$$\begin{aligned} r_\lambda^{\text{rheo}}(t - t') &= \frac{\lambda^{\text{ela}} \lambda_2^{\text{vis}}}{\lambda_1^{\text{vis}} + \lambda_2^{\text{vis}}} \exp \left\{ -\frac{\lambda^{\text{ela}}}{\lambda_1^{\text{vis}} + \lambda_2^{\text{vis}}} (t - t') \right\} \left(1 + \frac{\lambda_1^{\text{vis}}}{\lambda^{\text{ela}}} \frac{d}{dt} \right), \\ r_\mu^{\text{rheo}}(t - t') &= \frac{\mu^{\text{ela}} \mu_2^{\text{vis}}}{\mu_1^{\text{vis}} + \mu_2^{\text{vis}}} \exp \left\{ -\frac{\mu^{\text{ela}}}{\mu_1^{\text{vis}} + \mu_2^{\text{vis}}} (t - t') \right\} \left(1 + \frac{\mu_1^{\text{vis}}}{\mu^{\text{ela}}} \frac{d}{dt} \right). \end{aligned}$$

Two constants λ^{ela} and μ^{ela} specify elasticity of the object, λ_1^{vis} and μ_1^{vis} describe its viscosity, and λ_2^{vis} and μ_2^{vis} show its plasticity. Relaxation function $r_\lambda^{\text{rheo}}(t - t')$ characterizes normal viscoplasticity of the object while its shear viscoplasticity is described in $r_\mu^{\text{rheo}}(t - t')$.

The stress-strain relationship can be converted into a relationship between a set of forces applied to nodal points and a set of displacements of the points. As mentioned in Section 4.1.3, a set of elastic forces applied to nodal points is given by

$$\text{elastic force} = (\lambda J_\lambda + \mu J_\mu) \mathbf{u}_N,$$

where J_λ and J_μ are geometric matrices determined by object coordinate components of nodal points. The above equation suggests that replacing I_λ by J_λ , I_μ by J_μ , and $\boldsymbol{\varepsilon}$ by \mathbf{u}_N in the stress-strain relationship (5.11) of an elastic object yields the elastic force set.

From the above observation, replacing I_λ by J_λ , I_μ by J_μ , and $\boldsymbol{\varepsilon}$ by \mathbf{u}_N in the stress-strain relationship (5.13) of a viscoelastic object yields a set of viscoelastic forces applied to nodal points as follows:

$$\begin{aligned} \text{viscoelastic force} &= J_\lambda (\lambda^{\text{ela}} \mathbf{u}_N + \lambda^{\text{vis}} \dot{\mathbf{u}}_N) \\ &+ J_\mu (\mu^{\text{ela}} \mathbf{u}_N + \mu^{\text{vis}} \dot{\mathbf{u}}_N). \end{aligned} \quad (5.17)$$

Replacing I_λ by J_λ , I_μ by J_μ , and $\boldsymbol{\varepsilon}$ by \mathbf{u}_N in the stress-strain relationship (5.14) of a viscoplastic object yields a set of viscoelastic forces applied to

nodal points as follows:

$$\begin{aligned} \text{viscoplastic force} &= J_\lambda \int_0^t r_\lambda(t-t') \dot{\mathbf{u}}_N(t') dt' \\ &+ J_\mu \int_0^t r_\mu(t-t') \dot{\mathbf{u}}_N(t') dt'. \end{aligned} \quad (5.18)$$

Replacing r_λ by r_λ^{rheo} and r_μ by r_μ^{rheo} in the above equation yields a set of rheological forces applied to nodal points as follows:

$$\begin{aligned} \text{rheological force} &= J_\lambda \int_0^t r_\lambda^{\text{rheo}}(t-t') \dot{\mathbf{u}}_N(t') dt' \\ &+ J_\mu \int_0^t r_\mu^{\text{rheo}}(t-t') \dot{\mathbf{u}}_N(t') dt'. \end{aligned} \quad (5.19)$$

5.3.2 Computing 2D/3D viscoplastic deformation

Let us derive the dynamic equation of a 2D viscoplastic deformation. Let us introduce the following vectors:

$$\begin{aligned} \mathbf{w}_\lambda &= \int_0^t \lambda^{\text{ela}} \exp\left\{-\frac{\lambda^{\text{ela}}}{\lambda^{\text{vis}}}(t-t')\right\} \dot{\mathbf{u}}_N(t') dt', \\ \mathbf{w}_\mu &= \int_0^t \mu^{\text{ela}} \exp\left\{-\frac{\mu^{\text{ela}}}{\mu^{\text{vis}}}(t-t')\right\} \dot{\mathbf{u}}_N(t') dt'. \end{aligned}$$

Vectors \mathbf{w}_λ and \mathbf{w}_μ are referred to as *normal residual displacement vector* and *shear residual displacement vector*. A set of viscoplastic forces applied to nodal points is then simply described as

$$\text{viscoplastic force} = J_\lambda \mathbf{w}_\lambda + J_\mu \mathbf{w}_\mu.$$

Introduce the velocity vector $\mathbf{v}_N = \dot{\mathbf{u}}_N$ and replacing elastic force $K\mathbf{u}_N$ in eq.(4.18) by the viscoplastic force given in the above equation, a set of dynamic equations of nodal points is described as follows:

$$-(J_\lambda \mathbf{w}_\lambda + J_\mu \mathbf{w}_\mu) + \mathbf{f} + A\boldsymbol{\lambda} - M\dot{\mathbf{v}}_N = \mathbf{0},$$

where $\boldsymbol{\lambda}$ denotes a set of Lagrange multipliers corresponding to geometric constraints $A^T \mathbf{u}_N = \mathbf{0}$ and \mathbf{f} represents a set of external forces applied to the nodal points. Note that Lagrange multipliers describe the magnitude of

constraint forces corresponding to individual geometric constraints. Differentiating \mathbf{w}_λ and \mathbf{w}_μ with respect to time t , we have the following equations:

$$\begin{aligned}\dot{\mathbf{w}}_\lambda &= -\frac{\lambda^{\text{ela}}}{\lambda^{\text{vis}}}\mathbf{w}_\lambda + \lambda^{\text{ela}}\dot{\mathbf{u}}_N = -\frac{\lambda^{\text{ela}}}{\lambda^{\text{vis}}}\mathbf{w}_\lambda + \lambda^{\text{ela}}\mathbf{v}_N, \\ \dot{\mathbf{w}}_\mu &= -\frac{\mu^{\text{ela}}}{\mu^{\text{vis}}}\mathbf{w}_\mu + \mu^{\text{ela}}\dot{\mathbf{u}}_N = -\frac{\mu^{\text{ela}}}{\mu^{\text{vis}}}\mathbf{w}_\mu + \mu^{\text{ela}}\mathbf{v}_N.\end{aligned}$$

Applying the CSM to a set of geometric constraints $A^T\mathbf{u}_N = \mathbf{0}$ yields the following differential equation:

$$A^T\dot{\mathbf{v}}_N + A^T(2\omega\mathbf{v}_N + \omega^2\mathbf{u}_N) = \mathbf{0}.$$

Consequently, the equations of motion and the above differential equations can be described as follows:

$$\begin{aligned}\dot{\mathbf{u}}_N &= \mathbf{v}_N, \\ M\dot{\mathbf{v}}_N - A\boldsymbol{\lambda} &= -J_\lambda\mathbf{w}_\lambda - J_\mu\mathbf{w}_\mu + \mathbf{f}, \\ -A^T\dot{\mathbf{v}}_N &= A^T(2\omega\mathbf{v}_N + \omega^2\mathbf{u}_N), \\ \dot{\mathbf{w}}_\lambda &= -\frac{\lambda^{\text{ela}}}{\lambda^{\text{vis}}}\mathbf{w}_\lambda + \lambda^{\text{ela}}\mathbf{v}_N, \\ \dot{\mathbf{w}}_\mu &= -\frac{\mu^{\text{ela}}}{\mu^{\text{vis}}}\mathbf{w}_\mu + \mu^{\text{ela}}\mathbf{v}_N.\end{aligned}\tag{5.20}$$

Namely,

$$\begin{bmatrix} I & & & & & \\ & M & -A & & & \\ & -A^T & & & & \\ & & & I & & \\ & & & & I & \end{bmatrix} \begin{bmatrix} \dot{\mathbf{u}}_N \\ \dot{\mathbf{v}}_N \\ \boldsymbol{\lambda} \\ \dot{\mathbf{w}}_\lambda \\ \dot{\mathbf{w}}_\mu \end{bmatrix} = \begin{bmatrix} \mathbf{v}_N \\ -J_\lambda\mathbf{w}_\lambda - J_\mu\mathbf{w}_\mu + \mathbf{f} \\ A^T(2\omega\mathbf{v}_N + \omega^2\mathbf{u}_N) \\ -\frac{\lambda^{\text{ela}}}{\lambda^{\text{vis}}}\mathbf{w}_\lambda + \lambda^{\text{ela}}\mathbf{v}_N \\ -\frac{\mu^{\text{ela}}}{\mu^{\text{vis}}}\mathbf{w}_\mu + \mu^{\text{ela}}\mathbf{v}_N \end{bmatrix}.\tag{5.21}$$

Note that the above linear equation is solvable since the matrix is regular, implying that we can compute $\dot{\mathbf{u}}_N$, $\dot{\mathbf{v}}_N$, $\dot{\mathbf{w}}_\lambda$, and $\dot{\mathbf{w}}_\mu$. As a result, we can sketch \mathbf{u} , \mathbf{v} , \mathbf{w}_λ , and \mathbf{w}_μ using the Euler method or the Runge-Kutta method. In addition, as the matrix in the above equation is symmetric and block diagonal, we can compute its inverse matrix numerically in an efficient manner.

We can compute a set of viscoplastic forces given in $J_\lambda \mathbf{w}_\lambda + J_\mu \mathbf{w}_\mu$ without constructing total connection matrices J_λ and J_μ . Note that viscoplastic forces applied to nodal points P_i , P_j , and P_k caused by the deformation of triangle $T_p = \triangle P_i P_j P_k$ are approximated as follows:

$$\begin{bmatrix} \mathbf{f}_i^p \\ \mathbf{f}_j^p \\ \mathbf{f}_k^p \end{bmatrix} = J_\lambda^{i,j,k} \begin{bmatrix} \mathbf{w}_\lambda^i \\ \mathbf{w}_\lambda^j \\ \mathbf{w}_\lambda^k \end{bmatrix} + J_\mu^{i,j,k} \begin{bmatrix} \mathbf{w}_\mu^i \\ \mathbf{w}_\mu^j \\ \mathbf{w}_\mu^k \end{bmatrix},$$

where $J_\lambda^{i,j,k}$ and $J_\mu^{i,j,k}$ are partial connection matrices given in eqs.(4.10) and (4.11). Vectors \mathbf{w}_λ^i and \mathbf{w}_μ^i denote normal and shear residual displacement vectors at nodal point P_i . Summing up the contributions of all triangles to viscoplastic forces, we can obtain a set of viscoplastic forces applied to nodal points.

Example Let us demonstrate the formulation of 2D viscoplastic deformation by taking an example illustrated in Figure 4.6. A square shows isotropic viscoplasticity characterized by λ^{ela} , λ^{vis} , μ^{ela} , and μ^{vis} in its deformation. Displacement vector and velocity vector are described as

$$\mathbf{u}_N = \begin{bmatrix} \mathbf{u}_0 \\ \mathbf{u}_1 \\ \mathbf{u}_2 \\ \mathbf{u}_3 \\ \mathbf{u}_4 \\ \mathbf{u}_5 \end{bmatrix}, \quad \mathbf{v}_N = \begin{bmatrix} \mathbf{v}_0 \\ \mathbf{v}_1 \\ \mathbf{v}_2 \\ \mathbf{v}_3 \\ \mathbf{v}_4 \\ \mathbf{v}_5 \end{bmatrix}.$$

Normal and shear residual displacement vectors are described by

$$\mathbf{w}_\lambda = \begin{bmatrix} \mathbf{w}_\lambda^0 \\ \mathbf{w}_\lambda^1 \\ \mathbf{w}_\lambda^2 \\ \mathbf{w}_\lambda^3 \\ \mathbf{w}_\lambda^4 \\ \mathbf{w}_\lambda^5 \end{bmatrix}, \quad \mathbf{w}_\mu = \begin{bmatrix} \mathbf{w}_\mu^0 \\ \mathbf{w}_\mu^1 \\ \mathbf{w}_\mu^2 \\ \mathbf{w}_\mu^3 \\ \mathbf{w}_\mu^4 \\ \mathbf{w}_\mu^5 \end{bmatrix}.$$

Recall that partial connection matrices are given as

$$\begin{aligned}
 J_{\lambda}^{0,1,3} = J_{\lambda}^{1,2,5} &= \frac{1}{2} \left[\begin{array}{cc|cc|cc} 1 & 1 & -1 & 0 & 0 & -1 \\ 1 & 1 & -1 & 0 & 0 & -1 \\ \hline -1 & 0 & 1 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 & 0 & 1 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 1 & 1 & 0 & 1 \end{array} \right], \\
 J_{\mu}^{0,1,3} = J_{\mu}^{1,2,5} &= \frac{1}{2} \left[\begin{array}{cc|cc|cc} 3 & 1 & -2 & -1 & -1 & 0 \\ 1 & 3 & 0 & -1 & -1 & -2 \\ \hline -2 & 0 & 2 & 0 & 0 & 0 \\ -1 & -1 & 0 & 1 & 1 & 0 \\ \hline -1 & -1 & 0 & 1 & 1 & 0 \\ 0 & -2 & 0 & 0 & 0 & 2 \end{array} \right], \\
 J_{\lambda}^{1,4,3} = J_{\lambda}^{2,5,4} &= \frac{1}{2} \left[\begin{array}{cc|cc|cc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & -1 & 1 & 0 \\ \hline 0 & -1 & 1 & 1 & -1 & 0 \\ 0 & -1 & 1 & 1 & -1 & 0 \\ \hline 0 & 1 & -1 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array} \right], \\
 J_{\mu}^{1,4,3} = J_{\mu}^{2,5,4} &= \frac{1}{2} \left[\begin{array}{cc|cc|cc} 1 & 0 & -1 & -1 & 0 & 1 \\ 0 & 2 & 0 & -2 & 0 & 0 \\ \hline -1 & 0 & 3 & 1 & -2 & -1 \\ -1 & -2 & 1 & 3 & 0 & -1 \\ \hline 0 & 0 & -2 & 0 & 2 & 0 \\ 1 & 0 & -1 & -1 & 0 & 1 \end{array} \right].
 \end{aligned}$$

A set of viscoplastic forces applied to nodal points P_0 through P_5 is then formulated as follows:

$$\text{nodal viscoplastic forces} = \left[\begin{array}{cccc} \mathbf{f}_0^0 & & & \\ \mathbf{f}_1^0 & +\mathbf{f}_1^1 & +\mathbf{f}_1^2 & \\ & & \mathbf{f}_2^2 & +\mathbf{f}_2^3 \\ \mathbf{f}_3^0 & +\mathbf{f}_3^1 & & \\ & \mathbf{f}_4^1 & +\mathbf{f}_4^2 & +\mathbf{f}_4^3 \\ & & & \mathbf{f}_5^3 \end{array} \right],$$

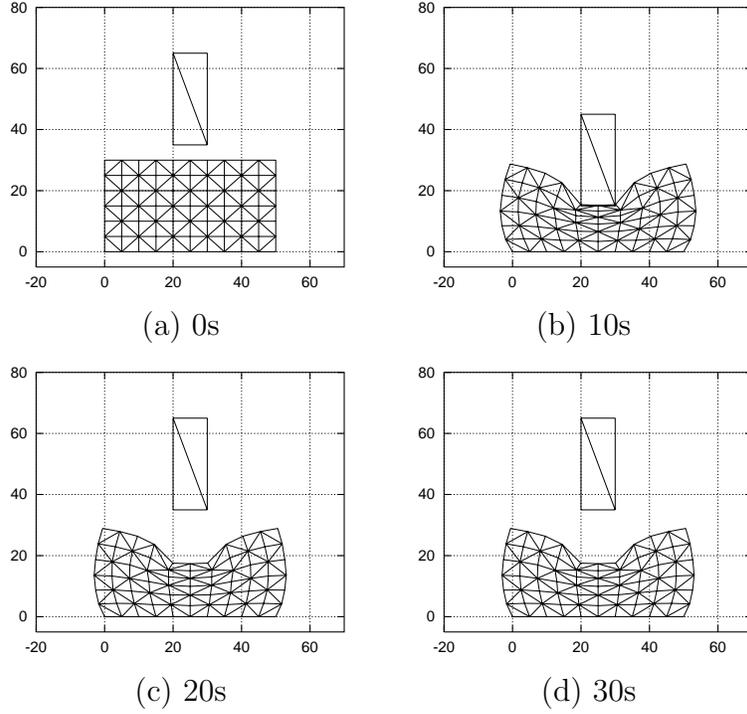


Figure 5.3: Computation of 2D viscoplastic deformation using FEM

where

$$\begin{aligned}
 \begin{bmatrix} \mathbf{f}_0^0 \\ \mathbf{f}_1^0 \\ \mathbf{f}_3^0 \end{bmatrix} &= J_\lambda^{0,1,3} \begin{bmatrix} \mathbf{w}_\lambda^0 \\ \mathbf{w}_\lambda^1 \\ \mathbf{w}_\lambda^3 \end{bmatrix} + J_\mu^{0,1,3} \begin{bmatrix} \mathbf{w}_\mu^0 \\ \mathbf{w}_\mu^1 \\ \mathbf{w}_\mu^3 \end{bmatrix}, \\
 \begin{bmatrix} \mathbf{f}_1^1 \\ \mathbf{f}_4^1 \\ \mathbf{f}_3^1 \end{bmatrix} &= J_\lambda^{1,4,3} \begin{bmatrix} \mathbf{w}_\lambda^1 \\ \mathbf{w}_\lambda^4 \\ \mathbf{w}_\lambda^3 \end{bmatrix} + J_\mu^{1,4,3} \begin{bmatrix} \mathbf{w}_\mu^1 \\ \mathbf{w}_\mu^4 \\ \mathbf{w}_\mu^3 \end{bmatrix}, \\
 \begin{bmatrix} \mathbf{f}_1^2 \\ \mathbf{f}_2^2 \\ \mathbf{f}_4^2 \end{bmatrix} &= J_\lambda^{1,2,4} \begin{bmatrix} \mathbf{w}_\lambda^1 \\ \mathbf{w}_\lambda^2 \\ \mathbf{w}_\lambda^4 \end{bmatrix} + J_\mu^{1,2,4} \begin{bmatrix} \mathbf{w}_\mu^1 \\ \mathbf{w}_\mu^2 \\ \mathbf{w}_\mu^4 \end{bmatrix}, \\
 \begin{bmatrix} \mathbf{f}_2^3 \\ \mathbf{f}_5^3 \\ \mathbf{f}_4^3 \end{bmatrix} &= J_\lambda^{2,5,4} \begin{bmatrix} \mathbf{w}_\lambda^2 \\ \mathbf{w}_\lambda^5 \\ \mathbf{w}_\lambda^4 \end{bmatrix} + J_\mu^{2,5,4} \begin{bmatrix} \mathbf{w}_\mu^2 \\ \mathbf{w}_\mu^5 \\ \mathbf{w}_\mu^4 \end{bmatrix}.
 \end{aligned}$$

Figure 5.3 demonstrates 2D Maxwell deformation. Elastic and viscous

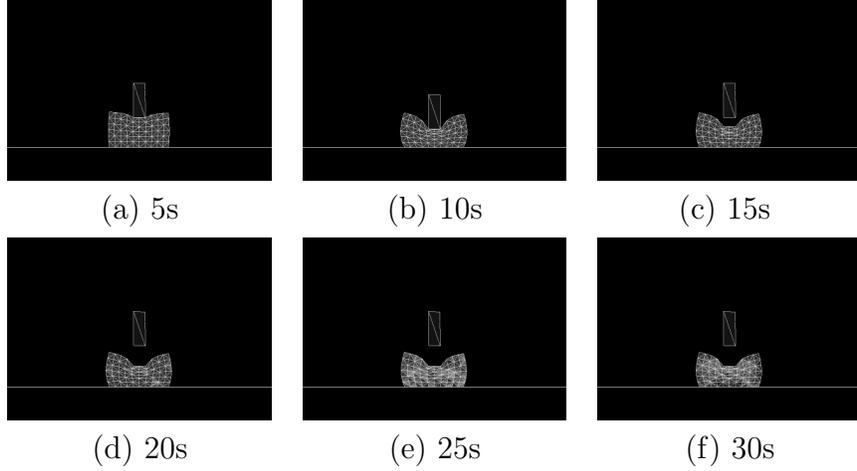


Figure 5.4: Computed stress in 2D viscoplastic deformation

moduli are $E = 300$ and $c = 500$. Poisson ratios for the moduli are given by $\nu^{\text{ela}} = 0.35$ and $\nu^{\text{vis}} = 0.35$. As shown in the figure, finite element approach can simulate viscoplastic deformation. Figure 5.4 denotes the stress imposed on individual triangles. Note that the stress is relaxed after the contact between the rigid bar and the elastic object is lost.

5.3.3 Computing 2D/3D rheological deformation

Let us derive the dynamic equation of a 2D rheological deformation. Let us introduce the following vectors:

$$\mathbf{w}_\lambda = \int_0^t \frac{\lambda^{\text{ela}} \lambda_2^{\text{vis}}}{\lambda_1^{\text{vis}} + \lambda_2^{\text{vis}}} e^{-\frac{\lambda^{\text{ela}}}{\lambda_1^{\text{vis}} + \lambda_2^{\text{vis}}}(t-t')} \left(\dot{\mathbf{u}}_N + \frac{\lambda_1^{\text{vis}}}{\lambda^{\text{ela}}} \ddot{\mathbf{u}}_N \right) (t') dt' - \frac{\lambda_1^{\text{vis}} \lambda_2^{\text{vis}}}{\lambda_1^{\text{vis}} + \lambda_2^{\text{vis}}} \dot{\mathbf{u}}_N,$$

$$\mathbf{w}_\mu = \int_0^t \frac{\mu^{\text{ela}} \mu_2^{\text{vis}}}{\mu_1^{\text{vis}} + \mu_2^{\text{vis}}} e^{-\frac{\mu^{\text{ela}}}{\mu_1^{\text{vis}} + \mu_2^{\text{vis}}}(t-t')} \left(\dot{\mathbf{u}}_N + \frac{\mu_1^{\text{vis}}}{\mu^{\text{ela}}} \ddot{\mathbf{u}}_N \right) (t') dt' - \frac{\mu_1^{\text{vis}} \mu_2^{\text{vis}}}{\mu_1^{\text{vis}} + \mu_2^{\text{vis}}} \dot{\mathbf{u}}_N.$$

A set of rheological forces applied to nodal points is then simply described as

$$\text{rheological force} = J_\lambda \left(\mathbf{w}_\lambda + \lambda^{\text{vis}} \dot{\mathbf{u}}_N \right) + J_\mu \left(\mathbf{w}_\mu + \mu^{\text{vis}} \dot{\mathbf{u}}_N \right).$$

where $\lambda^{\text{vis}} = \lambda_1^{\text{vis}} \lambda_2^{\text{vis}} / (\lambda_1^{\text{vis}} + \lambda_2^{\text{vis}})$ and $\mu^{\text{vis}} = \mu_1^{\text{vis}} \mu_2^{\text{vis}} / (\mu_1^{\text{vis}} + \mu_2^{\text{vis}})$. Introduce the velocity vector $\mathbf{v}_N = \dot{\mathbf{u}}_N$ and replacing elastic force $K\mathbf{u}_N$ in eq.(4.18) by

the rheological force given in the above equation, a set of dynamic equations of nodal points is described as follows:

$$- \left\{ J_\lambda \left(\mathbf{w}_\lambda + \lambda^{\text{vis}} \mathbf{v}_N \right) + J_\mu \left(\mathbf{w}_\mu + \mu^{\text{vis}} \mathbf{v}_N \right) \right\} + \mathbf{f} + A\boldsymbol{\lambda} - M\dot{\mathbf{v}}_N = \mathbf{0},$$

where $\boldsymbol{\lambda}$ denotes a set of Lagrange multipliers corresponding to geometric constraints $A^T \mathbf{u}_N = \mathbf{0}$ and \mathbf{f} represents a set of external forces applied to the nodal points. Differentiating \mathbf{w}_λ and \mathbf{w}_μ with respect to time t , we have the following equations:

$$\begin{aligned} \dot{\mathbf{w}}_\lambda &= -\frac{\lambda^{\text{ela}}}{\lambda_1^{\text{vis}} + \lambda_2^{\text{vis}}} (\mathbf{w}_\lambda - \lambda_2^{\text{vis}} \mathbf{v}_N), \\ \dot{\mathbf{w}}_\mu &= -\frac{\mu^{\text{ela}}}{\mu_1^{\text{vis}} + \mu_2^{\text{vis}}} (\mathbf{w}_\mu - \mu_2^{\text{vis}} \mathbf{v}_N). \end{aligned}$$

Applying the CSM to a set of geometric constraints $A^T \mathbf{u}_N = \mathbf{0}$ yields differential equations. Consequently, the equations of motion and the above differential equations can be described as follows:

$$\begin{aligned} \dot{\mathbf{u}}_N &= \mathbf{v}_N, \\ M\dot{\mathbf{v}}_N - A\boldsymbol{\lambda} &= -J_\lambda \left(\mathbf{w}_\lambda + \lambda^{\text{vis}} \mathbf{v}_N \right) - J_\mu \left(\mathbf{w}_\mu + \mu^{\text{vis}} \mathbf{v}_N \right) + \mathbf{f}, \\ -A^T \dot{\mathbf{v}}_N &= A^T (2\omega \mathbf{v}_N + \omega^2 \mathbf{u}_N), \\ \dot{\mathbf{w}}_\lambda &= -\frac{\lambda^{\text{ela}}}{\lambda_1^{\text{vis}} + \lambda_2^{\text{vis}}} (\mathbf{w}_\lambda - \lambda_2^{\text{vis}} \mathbf{v}_N), \\ \dot{\mathbf{w}}_\mu &= -\frac{\mu^{\text{ela}}}{\mu_1^{\text{vis}} + \mu_2^{\text{vis}}} (\mathbf{w}_\mu - \mu_2^{\text{vis}} \mathbf{v}_N). \end{aligned} \tag{5.22}$$

Namely,

$$\begin{bmatrix} I & & & & & \\ & M & -A & & & \\ & -A^T & & I & & \\ & & & & I & \\ & & & & & I \end{bmatrix} \begin{bmatrix} \dot{\mathbf{u}}_N \\ \dot{\mathbf{v}}_N \\ \boldsymbol{\lambda} \\ \dot{\mathbf{w}}_\lambda \\ \dot{\mathbf{w}}_\mu \end{bmatrix} = \begin{bmatrix} \mathbf{v}_N \\ -J_\lambda \left(\mathbf{w}_\lambda + \lambda^{\text{vis}} \mathbf{v}_N \right) - J_\mu \left(\mathbf{w}_\mu + \mu^{\text{vis}} \mathbf{v}_N \right) + \mathbf{f} \\ A^T (2\omega \mathbf{v}_N + \omega^2 \mathbf{u}_N) \\ -\frac{\lambda^{\text{ela}}}{\lambda_1^{\text{vis}} + \lambda_2^{\text{vis}}} (\mathbf{w}_\lambda - \lambda_2^{\text{vis}} \mathbf{v}_N) \\ -\frac{\mu^{\text{ela}}}{\mu_1^{\text{vis}} + \mu_2^{\text{vis}}} (\mathbf{w}_\mu - \mu_2^{\text{vis}} \mathbf{v}_N). \end{bmatrix}. \tag{5.23}$$

Note that the above linear equation is solvable since the matrix is regular, implying that we can compute $\dot{\mathbf{u}}_N$, $\dot{\mathbf{v}}_N$, $\dot{\mathbf{w}}_\lambda$, and $\dot{\mathbf{w}}_\mu$. As a result, we can sketch \mathbf{u} , \mathbf{v} , \mathbf{w}_λ , and \mathbf{w}_μ using the Euler method or the Runge-Kutta method. In addition, as the matrix in the above equation is symmetric and block diagonal, we can compute its inverse matrix numerically in an efficient manner.

We can compute a set of rheological forces given in $J_\lambda(\mathbf{w}_\lambda + \lambda^{\text{vis}}\mathbf{v}_N) + J_\mu(\mathbf{w}_\mu + \mu^{\text{vis}}\mathbf{v}_N)$ without constructing total connection matrices J_λ and J_μ . Note that rheological forces applied to nodal points P_i , P_j , and P_k caused by the deformation of triangle $T_p = \Delta P_i P_j P_k$ are approximated as follows:

$$\begin{bmatrix} \mathbf{f}_i^p \\ \mathbf{f}_j^p \\ \mathbf{f}_k^p \end{bmatrix} = J_\lambda^{i,j,k} \left(\begin{bmatrix} \mathbf{w}_\lambda^i \\ \mathbf{w}_\lambda^j \\ \mathbf{w}_\lambda^k \end{bmatrix} + \lambda^{\text{vis}} \begin{bmatrix} \mathbf{v}_i \\ \mathbf{v}_j \\ \mathbf{v}_k \end{bmatrix} \right) + J_\mu^{i,j,k} \left(\begin{bmatrix} \mathbf{w}_\mu^i \\ \mathbf{w}_\mu^j \\ \mathbf{w}_\mu^k \end{bmatrix} + \mu^{\text{vis}} \begin{bmatrix} \mathbf{v}_i \\ \mathbf{v}_j \\ \mathbf{v}_k \end{bmatrix} \right),$$

where $J_\lambda^{i,j,k}$ and $J_\mu^{i,j,k}$ are partial connection matrices given in eqs.(4.10) and (4.11). Summing up the contributions of all triangles to rheological forces, we can obtain a set of viscoplastic forces applied to nodal points.

Example Let us demonstrate the formulation of 2D rheological deformation by taking an example illustrated in Figure 4.6. A square shows isotropic rheological deformation characterized by λ^{ela} , λ_1^{vis} , λ_2^{vis} , μ^{ela} , μ_1^{vis} , and μ_2^{vis} . A set of viscoplastic forces applied to nodal points P_0 through P_5 is then formulated as follows:

$$\text{nodal rheological forces} = \begin{bmatrix} \mathbf{f}_0^0 \\ \mathbf{f}_1^0 + \mathbf{f}_1^1 + \mathbf{f}_1^2 \\ \mathbf{f}_2^2 + \mathbf{f}_2^3 \\ \mathbf{f}_3^0 + \mathbf{f}_3^1 \\ \mathbf{f}_4^1 + \mathbf{f}_4^2 + \mathbf{f}_4^3 \\ \mathbf{f}_5^3 \end{bmatrix},$$

where

$$\begin{bmatrix} \mathbf{f}_0^0 \\ \mathbf{f}_1^0 \\ \mathbf{f}_3^0 \end{bmatrix} = J_\lambda^{0,1,3} \left(\begin{bmatrix} \mathbf{w}_\lambda^0 \\ \mathbf{w}_\lambda^1 \\ \mathbf{w}_\lambda^3 \end{bmatrix} + \lambda^{\text{vis}} \begin{bmatrix} \mathbf{v}_0 \\ \mathbf{v}_1 \\ \mathbf{v}_3 \end{bmatrix} \right)$$

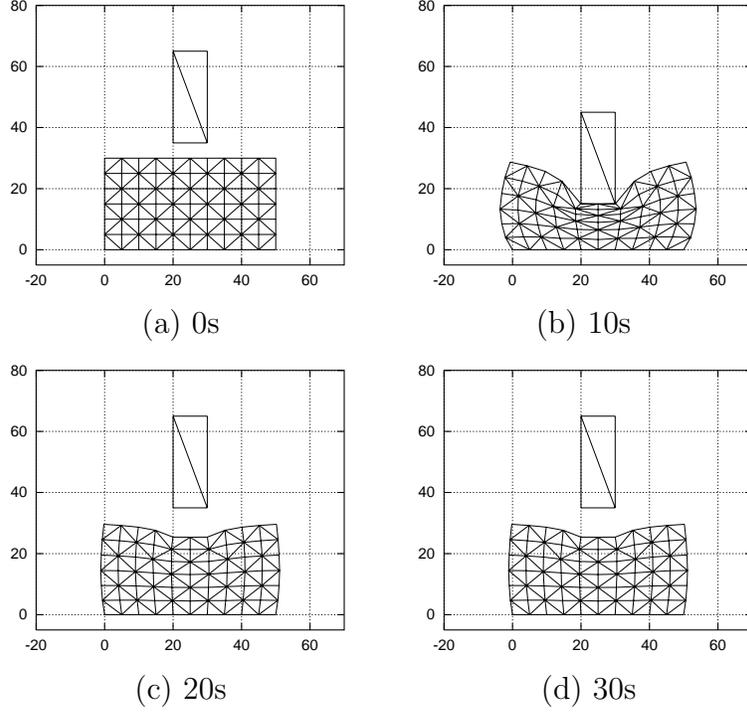


Figure 5.5: Computation of 2D rheological deformation using FEM

$$\begin{aligned}
 & + J_{\mu}^{0,1,3} \left(\begin{bmatrix} \mathbf{w}_{\mu}^0 \\ \mathbf{w}_{\mu}^1 \\ \mathbf{w}_{\mu}^3 \end{bmatrix} + \mu^{\text{vis}} \begin{bmatrix} \mathbf{v}_0 \\ \mathbf{v}_1 \\ \mathbf{v}_3 \end{bmatrix} \right), \\
 \begin{bmatrix} \mathbf{f}_1^1 \\ \mathbf{f}_4^1 \\ \mathbf{f}_3^1 \end{bmatrix} & = J_{\lambda}^{1,4,3} \left(\begin{bmatrix} \mathbf{w}_{\lambda}^1 \\ \mathbf{w}_{\lambda}^4 \\ \mathbf{w}_{\lambda}^3 \end{bmatrix} + \lambda^{\text{vis}} \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_4 \\ \mathbf{v}_3 \end{bmatrix} \right) \\
 & + J_{\mu}^{1,4,3} \left(\begin{bmatrix} \mathbf{w}_{\mu}^1 \\ \mathbf{w}_{\mu}^4 \\ \mathbf{w}_{\mu}^3 \end{bmatrix} + \mu^{\text{vis}} \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_4 \\ \mathbf{v}_3 \end{bmatrix} \right),
 \end{aligned}$$

and so on.

Figure 5.5 demonstrates 2D three-element deformation. Elastic and viscous moduli are $E = 30$, $c_1 = 20$, and $c_2 = 500$. Poisson ratios for the moduli are given by $\nu^{\text{ela}} = 0.35$, $\nu_1^{\text{vis}} = 0.35$, and $\nu_2^{\text{vis}} = 0.35$. As shown in the figure, finite element approach can simulate rheological deformation. Figure 5.6 denotes the stress imposed on individual triangles. Note that the

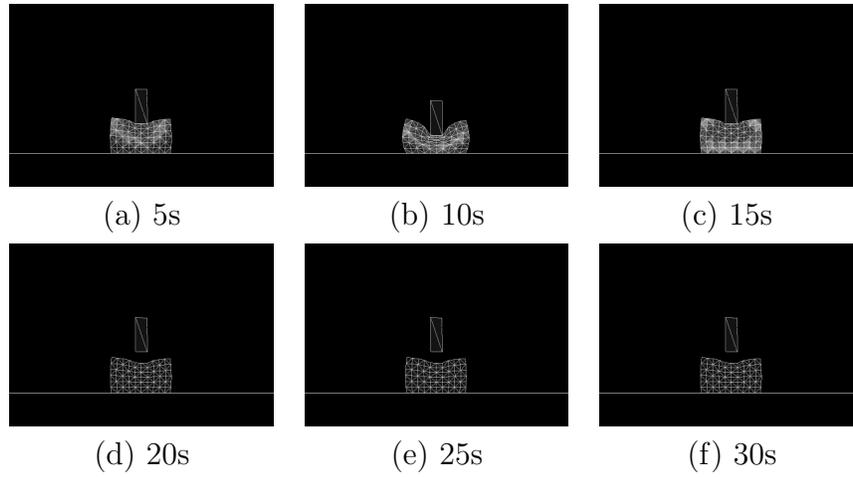


Figure 5.6: Computed stress in 2D rheological deformation

stress is relaxed after the contact between the rigid bar and the elastic object is lost.

第5章 仮想レオロジー物体構築のための一般化フォークトモデル

5.1 緒言

VR の分野において、柔軟物体のモデリングに関する研究が多くなされている。VR 上で柔軟物体を表現する際に三要素モデル、フォークトモデルが用いられている [25, 26]。しかし、現実にある柔軟物には、このようなモデルでは表現が不十分な場合が多い。実際のレオロジー物体のモデリングのために、仮想物体構築モデルの粘弾性要素に非線形的なパラメータを導入し、仮想物体を表現しようという試みがある。しかし、非線形要素は解析が困難であるため、実際の柔軟物体と仮想物体との挙動を合わせることは困難である。一方、線形要素はその運動解析が容易であり、仮に線形モデルで実物体の運動を表現できるならば、VR 上での仮想物体構築は容易である。物体変形において観察される非線形性の多くは、複数の線分の組合わせである折れ線で近似できることが多い。そこで、複数の線形要素を組み合わせて、非線形性を有する変形特性を表すことを試みる。本章では、フォークトモデルの組合わせである一般化フォークトモデルを用いて物体変形を表し、実際の変形計測から一般化フォークトモデルの力学パラメータを抽出する。

5.2 一般化フォークトモデル

本節では、仮想レオロジー物体モデリングに一般化フォークトモデル (generalized Voigt model) を導入する。低反発ウレタンスポンジ、および小麦粉生地を用いた変形実験の結果、フォークトモデルを用いたシミュレーション、および三要素モデルを用いたシミュレーションと実物体とで結果が大きく異なる部分が見受けられた。フォークトモデルでは荷重中に質点に作用する力が一定であるのに対し、低反発ウレタンスポンジでは力に減衰が見られる。同様の結果は、三要素モデルを用いたシミュレーション結果と小麦粉生地を用いた実験結果に見受けられる。この問題点を解決するために、一般化フォークトモデルを用いた仮想レオロジー物体の構築を試みる。

5.2.1 一般化フォークトモデル

一般化フォークトモデルは，フォークト要素を無質点で複数個直列に接続したモデルである．複数のフォークト要素が直列に接続されていることにより，フォークトモデルでは表し得ない複雑な挙動を表現することが可能となる．フォークトモデルは線形であり，その挙動は1つの減衰関数を用いて表現される．しかし，実際のレオロジー物体の挙動は線形的なものばかりでなく，非線形な挙動を示すものも多々存在する．そこで，一般化フォークトモデルを用いることにより，レオロジー物体の非線形的な挙動を線形の足し合わせで表現し，より実物体の挙動に近づけることができる．

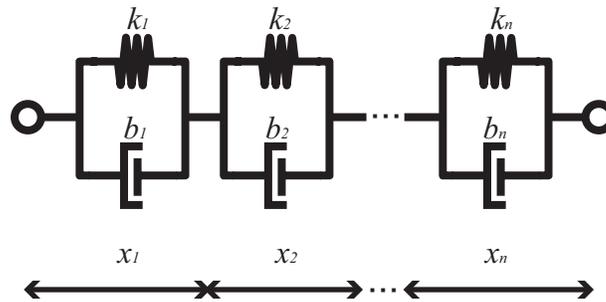


図 5.1: 一般化フォークトモデル

図 5.1 に示す n 要素一般化フォークトモデルにおける i 番目のフォークト要素の変位を Δx_i ，変位速度を $\Delta \dot{x}_i$ ，モデルの変位を Δx ，モデルの変位速度を $\Delta \dot{x}$ とすると， i 番目のフォークト要素の変位により生じる力 f は次式で表現できる．

$$f = -k_i \Delta x_i - b_i \Delta \dot{x}_i . \quad (5.1)$$

式 (5.1) を $\Delta \dot{x}_i$ について変形すると，次式を得る．

$$\Delta \dot{x}_i = \frac{-k_i \Delta x_i - f}{b_i} . \quad (5.2)$$

ここで， n 個の要素に関し，それぞれの要素の変位速度の総和をとることで次式を得る．

$$\sum_{i=1}^n \Delta \dot{x}_i = - \sum_{i=1}^n \frac{k_i}{b_i} \Delta x_i - f \sum_{i=1}^n \frac{1}{b_i} . \quad (5.3)$$

各要素の変位速度の総和は一般化フォークトモデルの質点の変位速度と等しいので，式 (5.3) を f について変形すると次式を得る．

$$f = \frac{- \sum_{i=1}^n \frac{k_i}{b_i} \Delta x_i - \Delta \dot{x}}{\sum_{i=1}^n \frac{1}{b_i}} . \quad (5.4)$$

式 (5.4) は，変形により一般化フックモデルの質点に作用する力をあらわしており，この値を式 (5.2) に代入することで，各要素の変位速度を求めることが可能となる．

5.2.2 内部摩擦考慮型粘性モデル

通常の粘性モデルでは，モデルに少しでも力が加わると変位速度が生じ，力が加わっている間変形し続けるという特徴がある．三要素モデルでは，荷重中にフックモデルの復元力が生じ，内部で構成要素の変形が生じている．そのため，フック要素は初期形状に復元するまで力を発生し続け，その結果粘性要素はフック要素が復元し終わるまで変形し続ける．つまり，三要素モデルでは荷重中に質点に作用する力は0へと収束するという結果となることが証明されている．しかし，実際のレオロジー物体を変形させた場合，荷重中に作用する力は0ではなく，ある一定の値へと収束していることが実験的にわかった．このようなシミュレーションモデルと実物体との違いは，三要素モデルの粘性要素に問題があると考え，要素の変形によりクーロン摩擦が生じるモデルを新たに提案する．

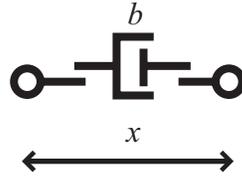


図 5.2: 内部摩擦を考慮した粘性モデル

図 5.2 に示す，内部摩擦考慮型粘性要素（以下，粘性摩擦要素）における変位速度を $\Delta\dot{x}$ ，変形により生じるクーロン摩擦力を f_{fric} とするとモデル変形により生じる力 f は次式のようになる．

$$f = -b\Delta\dot{x} + f_{fric} . \quad (5.5)$$

なお，クーロン摩擦力 f_{fric} の条件は次の通りである．

$$f_{fric} = \begin{cases} \alpha & (\Delta\dot{x} < 0) \\ 0 & (\Delta\dot{x} = 0) \\ -\alpha & (\Delta\dot{x} > 0) \end{cases} . \quad (5.6)$$

図 5.3 に，粘性摩擦モデルにおける変形速度と力の関係を示す．

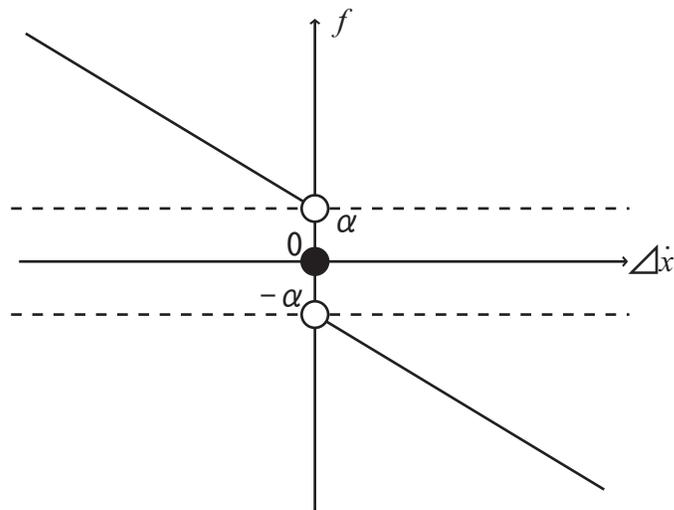


図 5.3: 粘性摩擦モデルにおける変形速度と力

5.3 変形シミュレーション

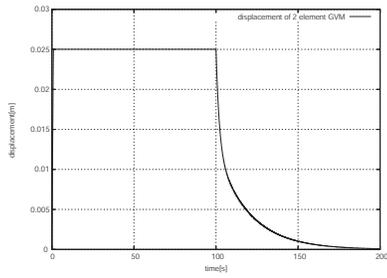
一般化フォークトモデルを用いて変形シミュレーションを行い，一般化フォークトモデルの特性を明確にする．特に，二要素，三要素の一般化フォークトモデルに対して，シミュレーションを行う．

5.3.1 二要素一般化フォークトモデルの変形シミュレーション

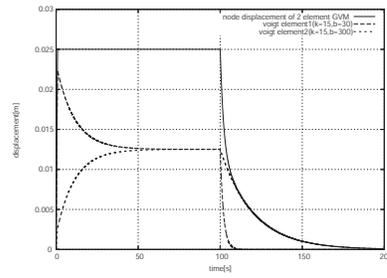
二要素一般化フォークトモデルの変形シミュレーションを行った．要素の粘弾性パラメータの異なるシミュレーションを2回行う．シミュレーションにおける各要素の粘弾性パラメータを，表 5.1 に示す．シミュレーションでは，一端を固定し，他端を 1s の間に一定速度で負方向に 0.025m 強制変位させ，圧縮する．次に，99s 間その状態を保持し，その後強制変位を解除する．シミュレーション結果を図 5.4 に示す．フォークトモデルと異なり，荷重中に応力緩和が見られる．

表 5.1: 二要素一般化フォークトモデルのシミュレーションパラメータ

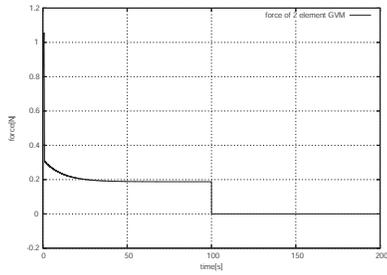
	k_1	b_1	k_2	b_2
simulation 1	15	30	15	300
simulation 2	15	3	15	300



(a) displacement

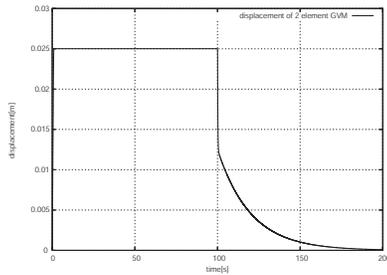


(b) displacement of each element

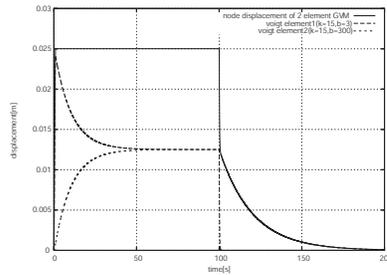


(c) force

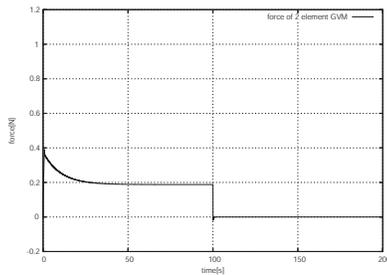
(a) simulation 1



(a) displacement



(b) displacement of each element



(c) force

(b) simulation 2

図 5.4: 二要素一般化フォクトモデルのシミュレーション結果

5.3.2 三要素一般化フォークトモデルの変形シミュレーション

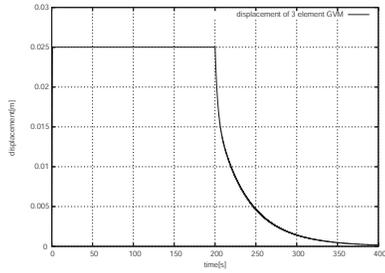
三要素一般化フォークトモデルの変形シミュレーションを行った。要素の粘弾性パラメータの異なるシミュレーションを2回行う。シミュレーションにおける各要素の粘弾性パラメータを、表5.2に示す。シミュレーションでは、一端を固定し、他端を1sの間に一定速度で負方向に0.025m強制変位させ、圧縮する。次に、変位後199s間その状態を保持し、その後強制変位を解除する。図5.5に三要素一般化フォークトモデルの変形シミュレーションの結果を示す。変位、力ともに二要素一般化フォークトとほぼ同じようなシミュレーション結果が見られる。

表 5.2: 三要素一般化フォークトモデルのシミュレーションパラメータ

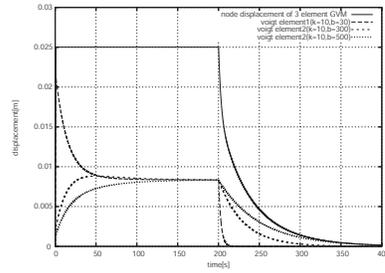
	k_1	b_1	k_2	b_2	k_3	b_3
simulation 1	10	30	10	300	10	500
simulation 2	10	60	10	300	10	600

5.3.3 内部摩擦考慮型三要素モデルの変形シミュレーション

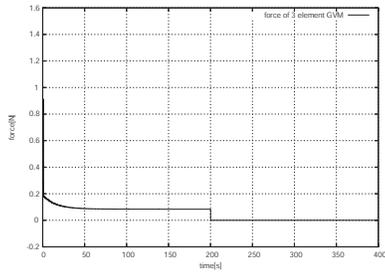
内部摩擦考慮型粘性要素を用いた三要素モデルの変形シミュレーションを行った。フォークト要素弾性係数 200N/m，フォークト要素粘性係数 100Ns/m，粘性要素粘性係数 1000Ns/m，内部摩擦の大きさ 2.0N である。一端を固定し、他端を1sの間に一定速度で負方向に0.025m強制変位させ、圧縮する。次に、19s間その状態を保持し、その後強制変位を解除する。シミュレーション結果を図5.6に示す。比較のため、内部摩擦の大きさを0Nとしたときの三要素モデル変形シミュレーションの結果を示す。図5.6(a)は変位、図5.6(b)は力を示す。図5.6(a)から内部摩擦を入れるときと入れないときとで変位が異なることがわかる。また、図5.6(b)を見ると、荷重中の力は内部摩擦を入れた場合0Nへと収束しているが、内部摩擦を入れた場合、内部摩擦と同じ値へと力が収束していることがわかる。シミュレーション結果より、レオロジー物体の表現するために内部摩擦考慮型粘性要素の導入することが望ましいと考えられる。



(a) displacement

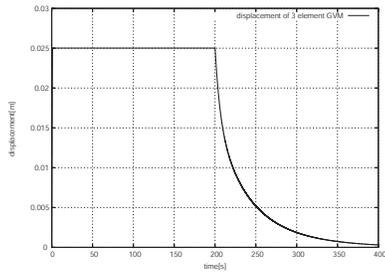


(b) displacement of each element

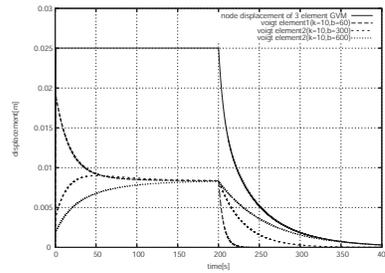


(c) force

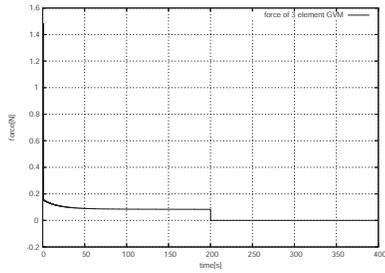
(a) simulation 1



(a) displacement



(b) displacement of each element



(c) force

(b) simulation 2

図 5.5: 三要素一般化フォクトモデルのシミュレーション結果

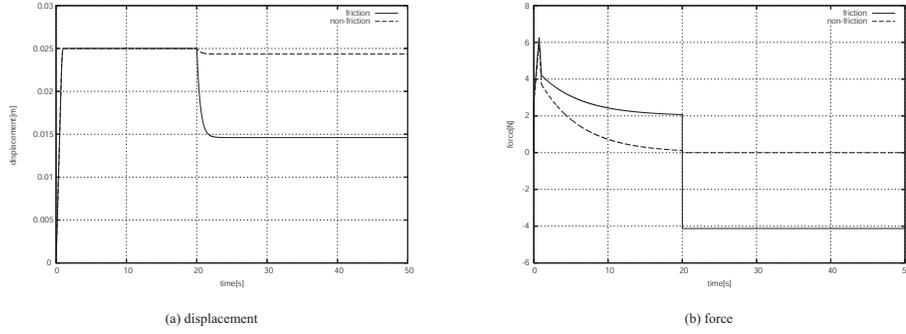


図 5.6: 内部摩擦考慮型三要素モデルのシミュレーション結果

5.4 力学解析とパラメータ推定

本章では，変形における一般化フォークトモデルの挙動を力学的に解析する．要素数 n の一般化フォークトモデルの変位は次の関係を満たす．

$$x = \sum_{i=1}^n x_i . \quad (5.7)$$

すなわち，一般化フォークトモデルを構成する各要素の変位の和がモデル自体の変位を示す．また， n 要素一般化フォークトモデルを構成する各フォークトモデルは，モデル自体の長さの $\frac{1}{n}$ の長さを持つようにし，モデルの質量を m とすると，各要素のもつ質量は $\frac{m}{n}$ であると仮定できる．このとき，除重後の一般化フォークトモデルにおける質点の運動は次式により定式化される．

$$x = \sum_{i=1}^n \frac{x_{iex}}{\lambda_{i2} - \lambda_{i1}} \left\{ \lambda_{i2} \exp(\lambda_{i1}(t - t_{ex})) - \lambda_{i1} \exp(\lambda_{i2}(t - t_{ex})) \right\} . \quad (5.8)$$

ただし，

$$\lambda_{i1} = -\frac{nb_i}{2m} + \frac{1}{2} \sqrt{\frac{b_i^2}{m^2} - \frac{4k_i}{m}} , \quad (5.9)$$

$$\lambda_{i2} = -\frac{nb_i}{2m} - \frac{1}{2} \sqrt{\frac{b_i^2}{m^2} - \frac{4k_i}{m}} . \quad (5.10)$$

ここで，三要素一般化フォークトモデルを例にとり，除重後の質点の運動を述べる．

図 5.7 は三要素一般化フォークトモデルの質点の変位の様子，および各構成要素の変位の様子を図示したものである．図 5.7 からわかるように，粘弾性パラメータの異なる各要素における変位の収束時間は異なる．除重直後は各要素は復元中であるため，モデル質点の運動は，構成する各フォークト要素の特徴を持つものとなが，時

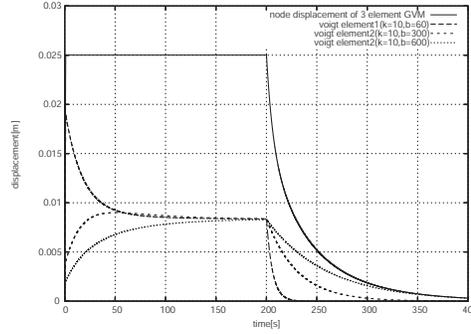


図 5.7: 三要素一般化フォークトモデルにおける各要素の変位

間の経過とともに復元力の大きいものから要素の復元が終わり，復元し終わった要素の特徴はモデル質点の挙動からなくなる．この性質から，一般化フォークトモデルの運動は除重後十分な時間経過すると，もっとも復元の遅い要素の特徴のみが現れるので，次式で表現できる．

$$x = \frac{x_{lateex}}{\lambda_{late2} - \lambda_{late1}} \left\{ \lambda_{late2} \exp(\lambda_{late1}(t - t_{ex})) - \lambda_{late1} \exp(\lambda_{late2}(t - t_{ex})) \right\} \quad (t \rightarrow \infty) . \quad (5.11)$$

最も復元の遅い要素の粘性係数 b_{late} は次式で求めることができる．

$$b_{late} = \frac{-k_{late} - \frac{m}{n} \lambda_{late1}^2}{\lambda_{late1}} . \quad (5.12)$$

式 (5.12) において， k_{late} は最も復元の遅い要素の弾性係数を示す．また，フォークトモデルの特徴から，除重後の戻り変位の自然体数値から，一般化フォークトモデルの構成要素数を推定することも可能である．フォークト要素の変位は近似的に次式で表すことができる．

$$x_{voigt} = A \exp(Bt) . \quad (5.13)$$

式 (5.7) と式 (5.13) から， n 要素一般化フォークトモデルは n 個の減衰関数の和で表現できる．

$$x = \sum_{i=1}^n A_i \exp(B_i t) . \quad (5.14)$$

二要素一般化フォークトモデルの場合，2つの減衰関数の和で表現できる．また，要素により減衰時間が異なるため，減衰の早い要素の特徴ほど先になくなる．そのため，時間の経過とともに現れる特徴は減少し，最終的には1要素の特徴しか現れない．したがって，図5.8に示すように，復元時の変位の自然体数値のグラフにおいて直線の数だけ要素があると考えられることができる．

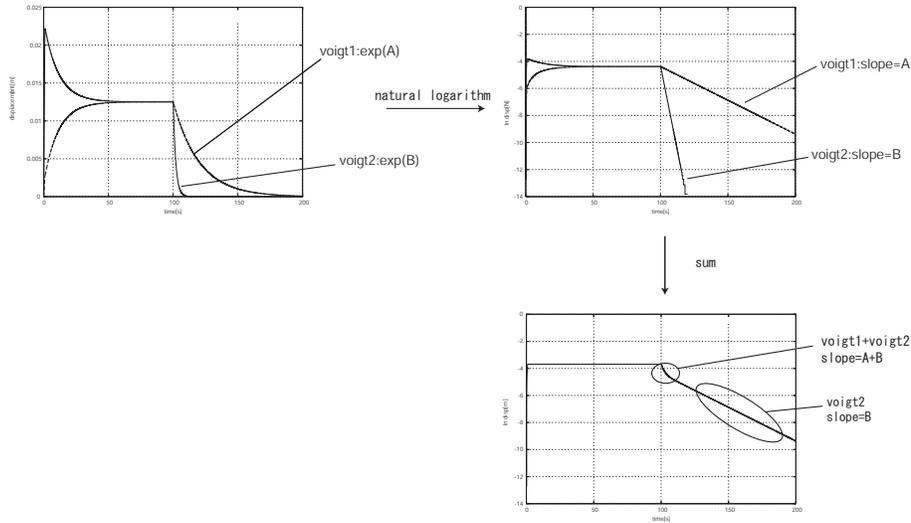


図 5.8: 一般化フォークトモデルにおける要素数の推定

5.4.1 荷重中に一般化フォークトモデルの質点に作用する力

荷重中，一般化フォークトモデルの質点に作用する力を定式化する．まず， n 要素一般化フォークトモデルにおける i 番目の要素が変形により発する力は次式のように表すことができる．

$$f(t) = -k_i \Delta x_i(t) - b_i \Delta \dot{x}_i(t) . \quad (5.15)$$

式 (5.15) は $\Delta x_i(t)$ に関する 1 階微分方程式であり， $\Delta x_i(t)$ について解くと次式を得る．

$$\Delta x_i(t) = \Delta x_{iex} \exp\left(-\frac{k_i}{b_i}t\right) + \int_{t_{ex}}^t \exp\left\{-\frac{k_i}{b_i}(t-\tau)\right\} \left\{-\frac{1}{b_i}f(\tau)\right\} d\tau . \quad (5.16)$$

式 (5.16) において x_{iex} は強制変位直後の i 番目の要素の変位を示す．また，式 (5.16) を微分することで以下の式を得る．

$$\begin{aligned} \Delta \dot{x}_i(t) &= -\frac{k_i}{b_i} x_{iex} \exp\left(-\frac{k_i}{b_i}t\right) + \frac{k_i}{b_i^2} \int_{t_{ex}}^t \exp\left\{-\frac{k_i}{b_i}(t-\tau)\right\} \{f(\tau)\} d\tau \\ &\quad - \frac{1}{b_i} f(t) . \end{aligned} \quad (5.17)$$

さらに

$$\Delta \ddot{x}_i(t) = \left(\frac{k_i}{b_i}\right)^2 x_{iex} \exp\left(-\frac{k_i}{b_i}t\right) - \frac{k_i^2}{b_i^3} \int_{t_{ex}}^t \exp\left\{-\frac{k_i}{b_i}(t-\tau)\right\} \{f(\tau)\} d\tau +$$

$$\frac{k_i}{b_i^2} f(t) - \frac{1}{b_i} \dot{f}(t). \quad (5.18)$$

モデルの変位 , およびモデル静止中の速度と加速度については以下の関係が成り立つ .

$$\Delta x = \sum_{i=1}^n \Delta x_i. \quad (5.19)$$

$$\begin{aligned} \Delta \dot{x} &= \sum_{i=1}^n \Delta \dot{x}_i \\ &= 0. \end{aligned} \quad (5.20)$$

$$\begin{aligned} \Delta \ddot{x} &= \sum_{i=1}^n \Delta \ddot{x}_i \\ &= 0. \end{aligned} \quad (5.21)$$

式 (5.18) を式 (5.21) に代入することで次式を得る .

$$\begin{aligned} A(t) &\stackrel{def}{=} \sum_{i=1}^n \left(\frac{k_i}{b_i} \right)^2 x_{iex} \exp \left(-\frac{k_i}{b_i} t \right) + \\ &\quad \sum_{i=1}^n -\frac{k_i^2}{b_i^3} \int_{t_{ex}}^t \exp \left\{ -\frac{k_i}{b_i} (t - \tau) \right\} \{f(\tau)\} d\tau + \\ &\quad \sum_{i=1}^n \frac{k_i}{b_i^2} f(t) + \sum_{i=1}^n -\frac{1}{b_i} \dot{f}(t) \\ &= 0. \end{aligned} \quad (5.22)$$

式 (5.22) の両辺にラプラス変換を施し , その結果のラプラス逆変換を求めることで静止中のモデルの力を定式化することが可能になる .

5.4.2 二要素一般化フォークトモデルの解析

二要素一般化フォークトモデルの場合 , 式 (5.22) において $n = 2$ とし , 両辺にラプラス変換を施すと次式を得る .

$$\begin{aligned} \mathcal{L}A(t) &= F(s) \left\{ \frac{-s^3 (b_1 + b_2) - s^2 (k_1 + k_2)}{(sb_1 + k_1)(sb_2 + k_2)} \right\} + \\ &\quad \frac{-s^2 (k_1 b_2 x_{1ex} + k_2 b_1 x_{2ex}) - s k_1 k_2 (x_{1ex} + x_{2ex})}{(sb_1 + k_1)(sb_2 + k_2)} \\ &= 0. \end{aligned} \quad (5.23)$$

式 (5.23) を $F(s)$ について解くと次のようになる .

$$F(s) = -\frac{s(k_1 b_2 x_{1ex} + k_2 b_1 x_{2ex}) + k_1 k_2 (x_{1ex} + x_{2ex})}{s \{s(b_1 + b_2) + (k_1 + k_2)\}}. \quad (5.24)$$

式 (5.24) を部分分数分解すると次式を得る .

$$F(s) = -\frac{1}{s} \frac{k_1 k_2}{k_1 + k_2} (x_{1ex} + x_{2ex}) + \frac{b_1 + b_2}{s(b_1 + b_2) + k_1 + k_2} \frac{(k_1 x_{1ex} - k_2 x_{2ex})(k_2 b_1 - k_1 b_2)}{(k_1 + k_2)(b_1 + b_2)}. \quad (5.25)$$

式 (5.25) の両辺の逆ラプラス変換を求めると次式を得る .

$$\begin{aligned} \mathcal{L}^{-1}F(s) &= f(t) \\ &= \frac{(k_1 x_{1ex} - k_2 x_{2ex})(k_2 b_1 - k_1 b_2)}{(k_1 + k_2)(b_1 + b_2)} \exp\left(-\frac{k_1 + k_2}{b_1 + b_2}t\right) - \\ &\quad \frac{k_1 k_2}{k_1 + k_2} (x_{1ex} + x_{2ex}). \end{aligned} \quad (5.26)$$

式 (5.26) は強制変位後の静止状態における 2 要素一般化フォークトモデルの質点に作用する力の様子を示す .

5.4.3 二要素一般化フォークトモデルの粘弾性パラメータ抽出

二要素一般化フォークトモデルにおける , 各要素の粘弾性パラメータの抽出法の提案する . まず , 求めるパラメータ数を減らすため , 構成要素の弾性係数は同じ値であると仮定する .

$$K = k_1 = k_2. \quad (5.27)$$

式 (5.26) より , 荷重中に二要素一般化フォークトモデルの質点に作用する力の収束値 f_{conv} は次式のようになる .

$$f_{conv} = -\frac{K}{2} (x_{1ex} + x_{2ex}). \quad (5.28)$$

ここで , $x_{1ex} + x_{2ex}$ はモデル自体の変位 Δx と等しいので , 式 (5.27) , 式 (5.28) より , 弾性係数は次式で求めることができる .

$$K = -\frac{2f_{conv}}{\Delta x}. \quad (5.29)$$

式 (5.29) と式 (5.12) から，各要素の粘弾性パラメータを求めることが可能である．ここで，シミュレーション結果から実際にパラメータの抽出を試みる．シミュレーションは以下の条件で行った．

- ディメンション：一次元
- シミュレーション時間：180s
- モデル質量：0.05kg
- モデル長さ：0.06m
- 片端固定し，6.1sの間に一定速度で負方向に0.045m強制変位させる（圧縮する）
- 変位後96.5sまでその状態を保持し，その後強制変位を解除

また，シミュレーションにおける各要素の粘弾性パラメータを，表 5.3 に示すように設定した．

表 5.3: 二要素一般化フォークトモデルのパラメータ設定

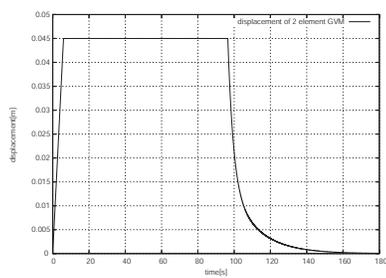
	k_1	b_1	k_2	b_2
parameter	10	30	20	300

シミュレーション結果を，図 5.9 に示す．図 5.9(c) において，荷重中に質点に作用する力の収束値は0.3Nであることが読み取れる．式 (5.29) より，弾性係数を $K = 13.333$ とする．また，図から，復元の遅い要素の粘性係数を求める．図 5.9(b) における160s以降の結果から表 5.4 のように値を読み取ると， $\lambda_{late1} = -0.0667$ となる．

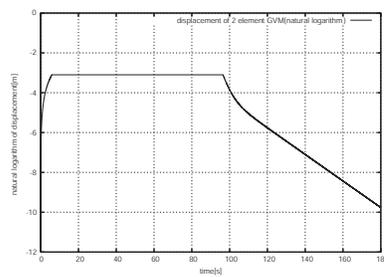
表 5.4: 二要素一般化フォークトモデルにおける変形の対数値

time[s]	natural logarithm value[m]
160	-8.43274
180	-9.76613

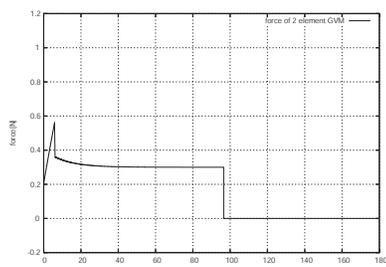
式 (5.12) から， $b_{late} = 196$ と推定できる．推定した要素のパラメータを用いて再度変形シミュレーションを行う．要素数 n の一般化フォークトモデルの場合，各要素の弾性係数が等しいとき，各要素はモデル自体の変位の $\frac{1}{n}$ に収束するように変位する．本シミュレーションの場合，モデル変位量は0.045mであるため，要素の強制変位量を0.0225mとしてシミュレーションを行う．



(a) displacement



(b) displacement (natural logarithmic value)



(c) force

図 5.9: 粘弾性パラメータ抽出のシミュレーション

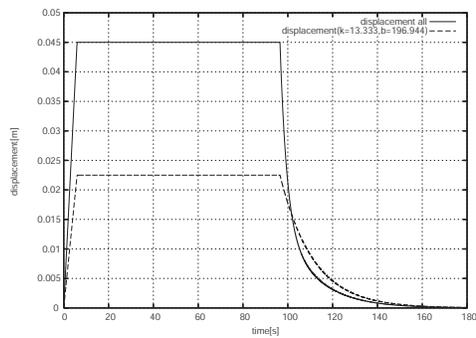


図 5.10: モデル質点の変位と推定値との比較 ($b = 196$)

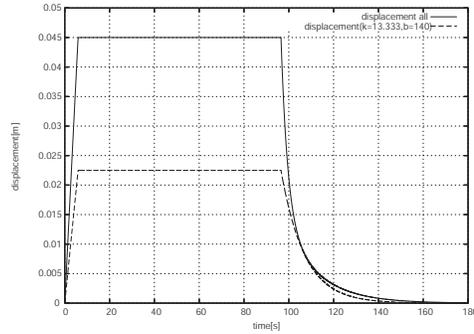


図 5.11: モデル質点の変位と推定値との比較 ($b = 140$)

図 5.10 を見ると、推定した要素の変位がモデル質点の変位を上回っている部分がある。モデル質点の変位から推定した要素の変位を減算できるように、推定要素の粘性係数を調整する。粘性係数を $b_{late} = 140$ としたときの要素の変位と比較すると図のようになる。

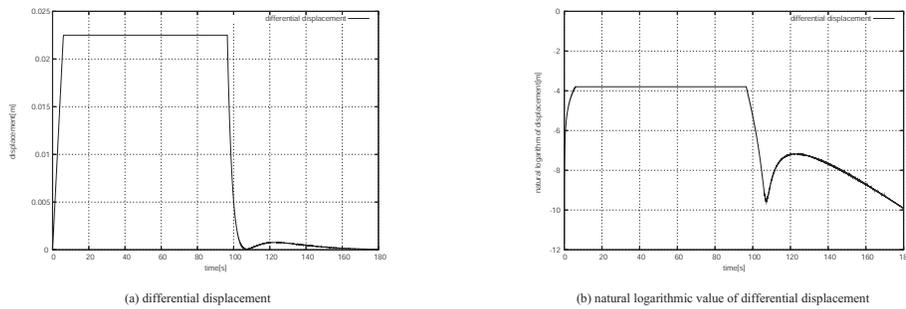


図 5.12: モデル質点の変位と推定値との差

図に示す状態になったら、モデル質点の変位から推定した要素の変位を引いた値とその自然対数値を求める。図 5.12(b) において、100s 前後の値から傾きを推定すると、 $\lambda_{late1} = -0.61$ となる。この結果からもうひとつの要素の粘性係数を推定すると、 $b = 22.0$ となる。

推定した 2 つの要素を用いた二要素一般化フォークトモデルの変位と力を元のデータと比較すると、図 5.13 が得られる。元のパラメータと推定したパラメータを表 5.5 に示す。

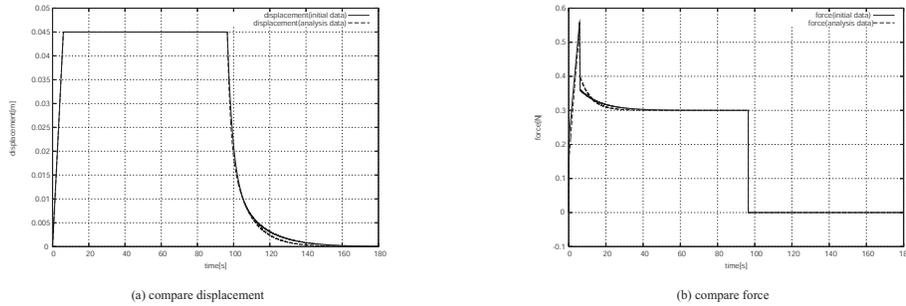


図 5.13: 実験値とシミュレーション値の比較

表 5.5: モデルパラメータと推定値

	K_1	b_1	k_2	b_2
initial	10	30	20	300
estimate	13.333	22.044	13.333	140

5.5 実験結果

5.5.1 低反発ウレタンスポンジの変形実験

本実験は、変形による残留変位がなく、かつ戻り変位速度が比較的ゆっくりである低反発ウレタンスポンジを用いて、実際の柔軟物体が変形によりどのような挙動を示すかを検証する。本実験において使用した装置を以下に示す。

- レーザ変位計 : LB-01 キーエンス製
- 超薄型圧力センサ : flexi force ニッタ製
- AD ボード : ADI12-16(PCI) CONTEC 製

図 5.14 に低反発ウレタン変形実験に用いた装置を示す。また、図 5.15 に実験のシステム図を示す。

本実験における手順を以下に述べる。

1. 低反発ウレタンスポンジ変形
2. 変形後一定時間形状を保持し、その間にウレタンに作用している力を flexi force にて計測

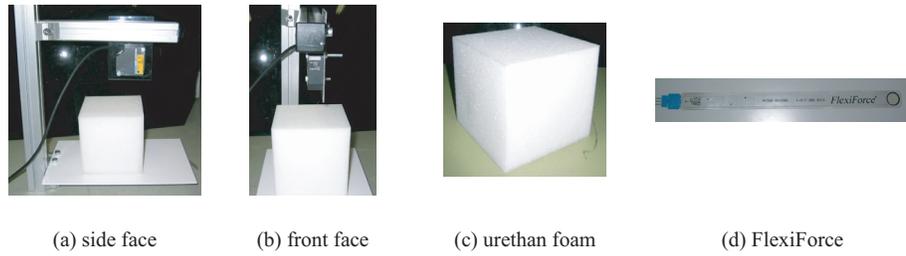


図 5.14: 低反発ウレタンの変形実験に用いた装置

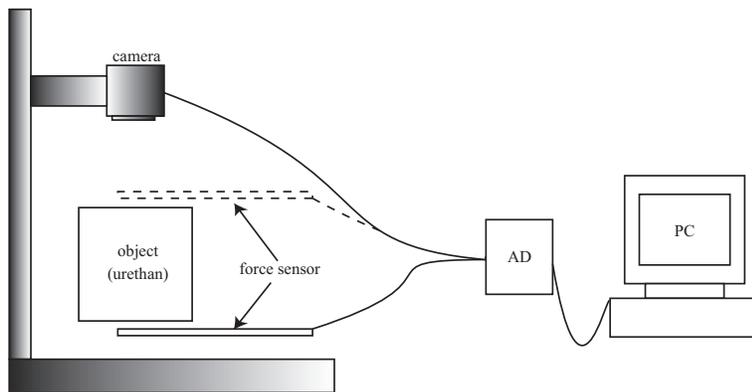


図 5.15: 低反発ウレタンの変形実験のシステム構成

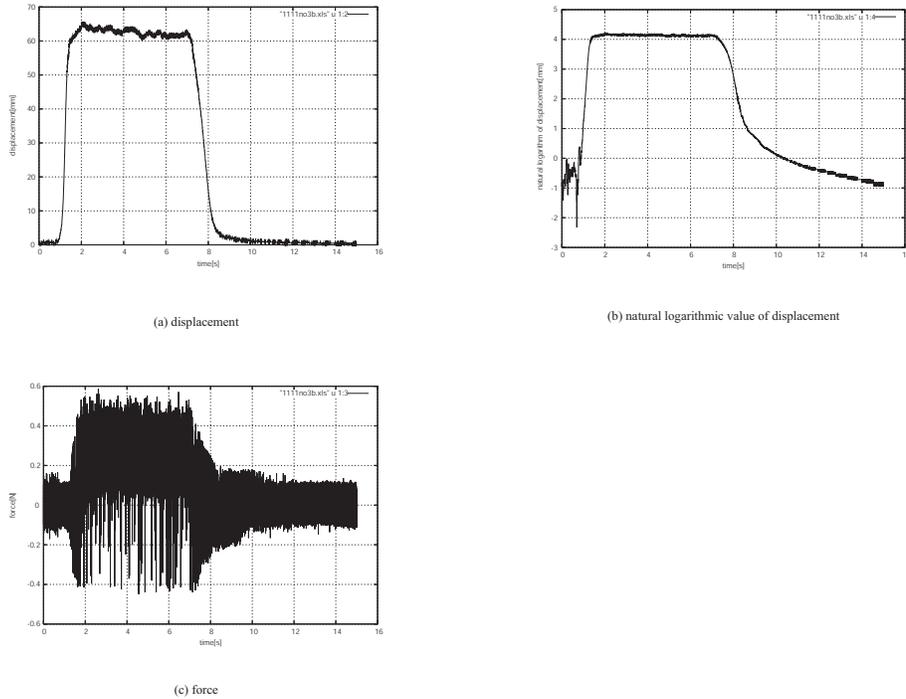


図 5.16: ウレタン底面における力の計測結果

3. 変形状態を解放し，低反発ウレタンスポンジの復元の様子をレーザ変位計により計測

低反発ウレタンスポンジの変形は，手で変形させた．また，レーザ変位計，および flexi force による計測結果は AD ボードによりデジタル変換される．なお，AD ボードのサンプリングタイムは 1ms である．圧力センサによる力の計測はウレタンの底面，および上面について別々に計測し，それぞれについて，レーザ変位計との同期を取る．

実験の結果を示す．図 5.16 は変形によりウレタン底面にかかる力を計測した結果，図 5.17 は上面にかかる力を計測した結果である．図 5.16(b) および図 5.17(b) より，低反発ウレタンスポンジの変位の自然体数値は直線的ではなく，フォークトモデルの変形結果と大きく異なることがわかる．また，図 5.16(c) および図 5.17(c) より，ウレタンに作用する力は減衰傾向にあり，応力緩和が見られる．フォークトモデルでは応力緩和が見られないことから，変位の計測結果，および力の計測結果から低反発ウレタンスポンジはフォークトモデルでは表現できないことがわかる．

また，図 5.16(c)，および図 5.17(c) からわかるように，圧力センサのノイズが大きいので，立命館大学情報理工学部田中弘美研究室において，ロードセルを用いた

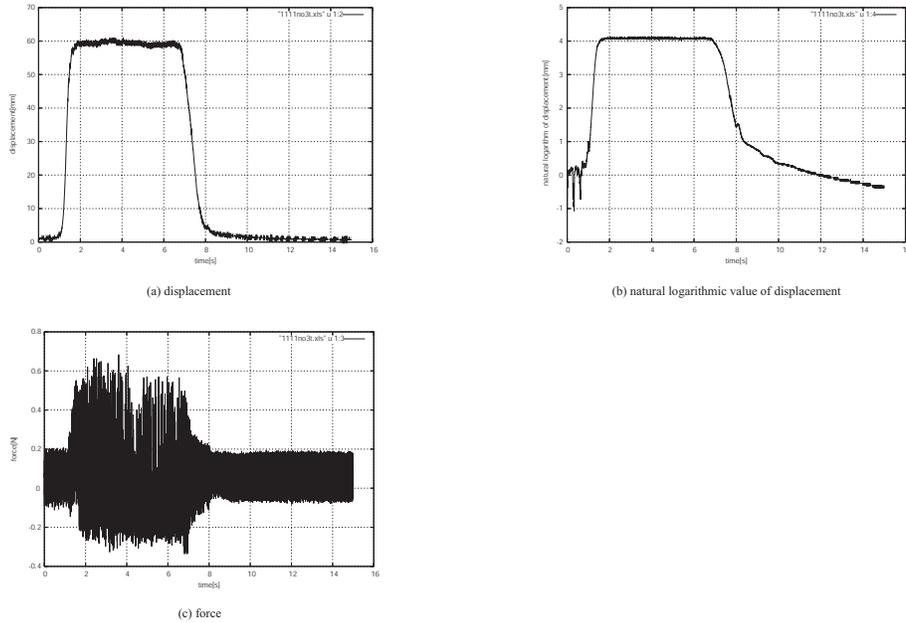


図 5.17: ウレタン上面における力の計測結果

ウレタン上面に作用する力の計測を行った．ロードセルによる計測は以下の条件で行った．

- サンプルングタイム：0.1s
- シミュレーション時間：180s
- 6.1s の間に一定速度で負方向に 0.045m 強制変位させる（圧縮する）
- 変位後 96.5s までその状態を保持し，その後強制変位を解除

ロードセルによる力の計測結果を図 5.18 に示す．図 5.18 においても，変形により低反発ウレタンスポンジに作用する力は減衰傾向にあることがわかった．この結果からも，低反発ウレタンスポンジの変形はフォークトモデルでは表現できず，弾性物体の挙動を表現するにはフォークトモデルに代わるモデルを用いる必要がある．

5.5.2 実験結果とシミュレーションの比較

一般化フォークトモデルを用いて低反発ウレタンスポンジを用いた実験結果から粘弾性パラメータを抽出し，シミュレーションと実験結果とを比較する．

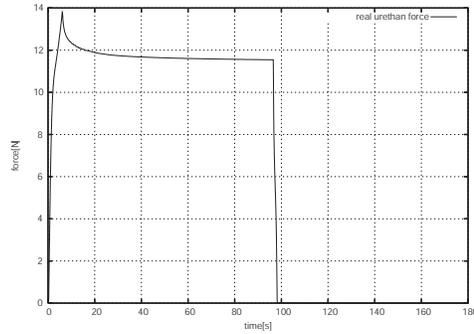
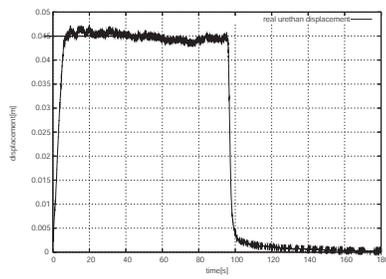


図 5.18: ロードセルによる力の計測結果

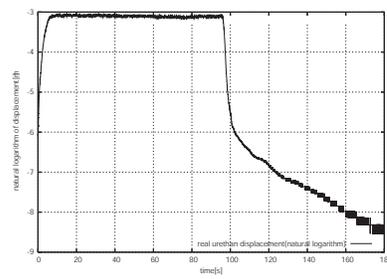
まず，図 5.19(b) より，実験に用いた低反発ウレタンスポンジが，二要素一般化フォークトモデルで表現できると仮定する．次に，図 5.19(c) より，変形により低反発ウレタンスポンジに作用する力は $11.55616N$ に収束していることが読み取れる．その値と，式 (5.29) より，一般化フォークトモデル構成要素の弾性係数は $K = 513.607$ と推定できる．また，図 5.19(b) において， $105s$ 以降のデータから， λ_{late1} を求めると， $\lambda_{late1} = -0.0327$ である．この値と式 (5.12) から $b_{late} = 15706.637$ と推定できる．第 5.4.3 節で述べたように， b_{late} の値を調整していった結果， $b_{late} = 1000$ という推定値を得た．

推定パラメータを用いたフォークト要素の変形結果を，低反発ウレタンスポンジ変形実験結果から引いた値，およびそのその自然体数値を図 5.20 に示す．

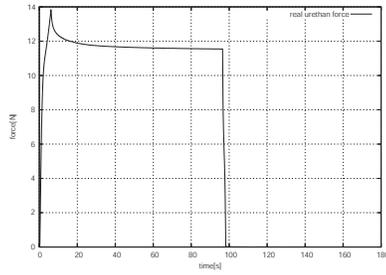
図 5.20(b) において， $100s$ 前後の値から，再び λ_1 の値を推定すると， $\lambda_1 = -2.16$ となった．式 (5.12) より，もうひとつの要素の粘性係数を推定すると， $b = 237$ という値が得られた．図 5.21 に実験結果と推定したパラメータを用いたシミュレーション結果の変位と作用力の比較を示す．変位，作用力ともに，大きなずれはなく，この結果から，低反発ウレタンスポンジは二要素一般化フォークトモデルを用いて表現することができる．



(a) displacement

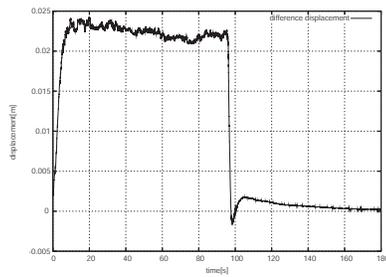


(b) natural logarithmic value of displacement

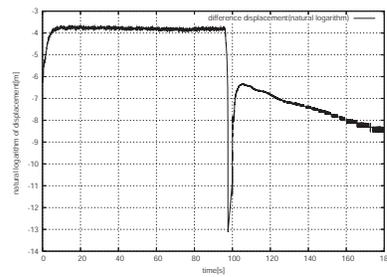


(c) force

図 5.19: 低反発ウレタンスポンジの変形実験の結果

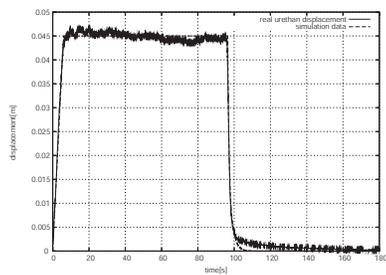


(a) differential displacement

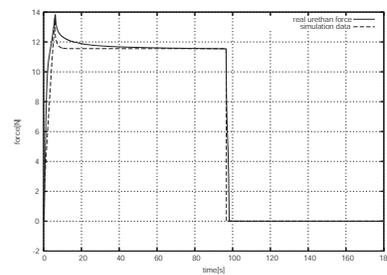


(b) natural logarithmic value of differential displacement

図 5.20: 変形実験の結果と推定値との差



(a) compare displacement



(b) compare force

図 5.21: 実験値とシミュレーション値の比較

参考文献

- [1] J. Baumgarte, *Stabilization of Constraints and Integrals of Motion in Dynamical Systems*, Computer Methods in Applied Mechanics and Engineering, Vol. 1, pp.1–16 (1972).
- [2] D. Breen, D. House, and M. Wozny, *Predicting the Drape of Woven Cloth Using Interacting Particles*, Computer Graphics(Proc. SIGGRAPH'94), pp.365–372 (1994).
- [3] S. Cotin, H. Delingette, and N. Ayache, *A Hybrid Elastic Model allowing Real-Time Cutting, Deformations and Force-Feedback for Surgery Training and Simulation*, Visual Computer, Vol.16, No.8, pp.437–452 (2000).
- [4] G. Debunne, M. Desbrun, M.-P. Cani, and A. H. Barr, *Dynamic Real-Time Deformations using Space & Time Adaptive Sampling*, Computer Graphics Proceedings (SIGGRAPH 2001), pp.31–36 (2001).
- [5] B. Eberhardt, A. Weber, and W. Strasser, *A Fast, Flexible, Particle-System Model for Cloth Draping*, IEEE Computer Graphics and Applications, pp.52–59 (1996).
- [6] Makino Junichiro, Fukushige Toshiyuki, Koga Masaki, and Namura Ken, *GRAPE-6: The massively-parallel special-purpose computer for astrophysical particle simulation*, PASJ, Vol.55, No.6 (2003).
- [7] 広田光一, 金子豊久, 仮想物体の弾性モデルに関する検討, 計測自動制御学会論文集, Vol.32, No.3, pp.232–238 (1998).
- [8] D. L. James and D. K. Pai, *ArtDefo Accurate Real Time Deformable Objects*, Computer Graphics Proceedings (SIGGRAPH 1999), pp.65–72 (1999).
- [9] D. L. James and D. K. Pai, *Real Time Simulation of Multizone Elastokinematic Models*, Proc. IEEE Int. Conf. Robotics and Automation, pp.927–32, (2002).

- [10] H. Keller, H. Stolz, A. Ziegler, and T. Braunl, *Virtual Mechanics Simulation and Animation of Rigid Body System*, <http://citeseer.nj.nec.com/keller94virtual.html>, p.28 (1993).
- [11] 増淵雄一, 分子レオロジーシミュレータ NAPLES, <http://masubuchi.jp/NAPLESWEB/>.
- [12] 宮崎慎也, 吉田俊介, 安田孝美, 横井茂樹, 局所形状保持に基づく仮想弾性物体モデルの提案, 電子情報通信学会論文誌 A, Vol.J82-A, No.7, pp.1148–1155 (1999).
- [13] M. Moore and J. Wilhelms, *Collision Detection and Response for Computer Animation*, Computer Graphics, Vol.22, No.4, pp.289–298 (1988).
- [14] 日本レオロジー学会編, 講座・レオロジー, 高分子刊行会 (1992).
- [15] G. Picinbono, H. Delingette, and N. Ayache *Non-linear and anisotropic elastic soft tissue models for medical simulation*, Proc. 2001 IEEE Int. Conf. on Robotics and Automation, Seoul, pp.1371–1376 (2001).
- [16] W. Reeves, *Particle Systems – A Technique for Modeling a Class of Fuzzy Objects*, Computer Graphics, Vol.2, No.2, pp.91–108 (1983).
- [17] Katsuhito Akahane, Shoichi Hasegawa, Yasuharu Koike, and Makoto Sato, 10kHz の更新周波数を実現する高解像度ハプティックコントローラの開発, 日本バーチャルリアリティ学会誌, (2004).
- [18] 種谷真一, やさしいレオロジー工学, 工業調査会 (1990).
- [19] H. Tanaka and F. Kishino, *Adaptive Mesh Generations for Surface Reconstruction: Parallel Hierarchical Triangulation without Cracks*, Proc. IEEE Computer Vision and Pattern Recognition, pp.88–94 (1993).
- [20] D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer, *Elastically Deformable Models*, Computer Graphics, Vol.21, No4, pp.205–214 (1987).
- [21] D. Terzopoulos and A. Witkin, *Deformable Models Physically Based Models with Rigid and Deformable Components*, IEEE Computer Graphics and Applications, November, pp.41–51 (1988).

- [22] D. Terzopoulos and K. Fleischer, *Modeling Inelastic Deformation: Viscoelasticity, Plasticity, Fracture*, Computer Graphics, Vol.22, No.4, pp.269–278 (1988).
- [23] D. Terzopoulos and K. Waters, *Physically-based facial modeling, analysis, and animation*, Visualization and Computer Animation, Vol.1, No.2, pp.73–80 (1990).
- [24] 徳本真一, 平井慎一, 形状制御のためのレオロジー物体の変形モデリング, 日本ロボット学会誌, Vol.18, No.8, pp.87–94 (2001).
- [25] 友國誠至, 杉山勇太, 平井慎一, 実時間計算可能な仮想レオロジー物体の構築, 日本バーチャルリアリティ学会論文誌, Vol.8, No.3, pp.247–254 (2003).
- [26] 植田直樹, 榎浜斎延, 平井慎一, 田中弘美, ハプティックビジョンに基づく能動的レオロジー物体モデリングのための粘弾性パラメータ抽出, 日本バーチャルリアリティ学会論文誌, Vol.8, No.3, pp.237–246 (2003).
- [27] Y. Zhuang and J. Canny, *Haptic Interaction with Global Deformations*, Proc. 2000 IEEE Int. Conf. on Robotics and Automation, San Francisco, pp.2428–2433 (2000).

成果

- [1] 友國誠至, 平井慎一, FPGA による仮想レオロジー物体のリアルタイム変形シミュレーション, 日本バーチャルリアリティ学会論文誌, 投稿中.
- [2] 友國誠至, 平井慎一, FPGA による仮想レオロジー物体のリアルタイム変形計算, 計測自動制御学会システムインテグレーション部門学術講演会, pp.979–980, 2004.
- [3] Ryo Nogami, Hiroshi Noborio, Seiji Tomokuni, and Shinichi Hirai, *A Comparative Study of Rheology MSD Models whose Structures are Lattice and Truss*, Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp.3809–3616, Sendai, September, 2004.
- [4] Naoki Ueda, Shinichi Hirai, and Hiromi T.Tanaka, *Extracting Rheological Properties of Deformable Objects with Haptic Vision*, Proc. IEEE Int. Conf. on Robotics and Automation, pp.3902–3907, New Orleans, April, 2004.
- [5] Ryo Nogami, Fumiaki Ujibe, Hiroki Fujii and Hiroshi Noborio, *Precise Deformation of Rheologic Object under MSD Models with Many Voxels and Calibrating Parameters*, Proc. IEEE Int. Conf. on Robotics and Automation, pp.1919–1926, New Orleans, April, 2004.
- [6] 平井慎一, 友國誠至, レオロジー変形の動的モデリング, ロボティクス・メカトロニクス'04 講演会予稿集 CD-ROM, 2004.
- [7] Yuuta Sugiyama and Shinichi Hirai, *Crawling and Jumping of Deformable Soft Robot*, Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp.3276–3281, Sendai, September, 2004.
- [8] Yuuta Sugiyama and Shinichi Hirai, *Crawling and Jumping by a Deformable Robot*, Proc. Int. Symp. on Experimental Robotics, Singapore, June, 2004.

- [9] Anand Vaz and Shinichi Hirai, *A Simplified Model for a Biomechanical Joint with Soft Cartilage*, Proc. 2004 IEEE Int. Conf. on Systems, Man, and Cybernetics (SMC 2004), paper no.1063, The Hague, The Netherlands, Oct., 2004.
- [10] Shinichi Hirai, *Indirect Simultaneous Positioning of Deformable Objects by Redundant Fingers without Physical Parameters*, Complex Systems Intelligence and Modern Technological Applications (CSIMTA 2004), pp.509-514, Cherbourg, France, September, 2004.
- [11] Shinichi Hirai, *Indirect Simultaneous Positioning of Deformable Objects without Physical Parameters and Time-Derivatives*, Int. Conf. on Machine Automation (ICMA2004), pp.69-74, Osaka, Japan, November, 2004.
- [12] Hiroshi Noborio, Ryo Nogami, Ryo Enoki, *Precise Deformation of Rheology MSD Model Calibrated by Randomized Algorithm*, Eurographics 2003, Granada Spain, pp.171-178, 2003.
- [13] 友國誠至, 杉山勇太, 平井慎一, 実時間計算可能な仮想レオロジー物体の構築, 日本バーチャルリアリティ学会論文誌, Vol.8, No.3, pp.247-254, 2003.
- [14] 植田直樹, 榑浜斎延, 平井慎一, 田中弘美, ハプティックビジョンに基づく能動的レオロジー物体モデリングのための粘弾性パラメータ抽出, 日本バーチャルリアリティ学会論文誌, Vol.8, No.3, pp.237-246, 2003.
- [15] 野上良, 榎亮, 登尾啓史, レオロジー物体を表現する3つのモデルの変形特性に関する研究, 日本バーチャルリアリティ学会論文集, Vol.8, No.3, pp.271-278, 2003.
- [16] 友國誠至, 杉山勇太, 木村政文, 平井慎一, 仮想レオロジー物体のモデリングおよびFPGAによる実時間変形計算, ロボティクス・メカトロニクス'03 講演会予稿集 CD-ROM, 2003.
- [17] Masafumi Kimura, Yuuta Sugiyama, Seiji Tomokuni, and Shinichi Hirai, *Constructing Rheologically Deformable Virtual Objects*, Proc. IEEE Int. Conf. on Robotics and Automation, pp.3737-3743, Taipei, September, 2003.
- [18] Hiroshi Noborio, Ryo Enoki, Shohei Nishimoto and Takumi Tanemura, *On the Calibration of Deformation Model of Rheology Object by a Modified Randomized Algorithm*, Proc. IEEE Int. Conf. on Robotics and Automation, pp.3729-3736, Taipei, September, 2003.

- [19] 杉山勇太, 木村政文, 友國誠至, 平井慎一, 仮想レオロジー物体における接触のモデリング, 計測自動制御学会システムインテグレーション部門学術講演会予稿集, Vol.3, pp.23-24, 2002.
- [20] 友國誠至, 木村 政文, 杉山 勇太, 平井 慎一, 三要素モデルによる粘弾性物体, 塑性物体およびレオロジー物体のモデリング, 計測自動制御学会システムインテグレーション部門学術講演会予稿集, Vol.3, pp.25-26, 2002.
- [21] 木村政文, 杉山勇太, 友國誠至, 平井慎一, 仮想レオロジー物体の変形シミュレーション, 日本バーチャルリアリティ学会第7回大会論文集 CD-ROM, 2002.