基礎論文

TVRSJ Vol.10 No.3, 2005

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

友國　誠至[*1]　　平井　慎一[*2]

Real-time Simulation of Rheological Deformation on FPGA

Seiji Tomokuni[*1], and Shinichi Hirai[*2]

Abstract – **Deformable soft objects such as food and tissue show both elastic and viscoplastic properties, and are referred to as rheological objects. A physical model of virtual objects has been developed, but computer power is insufficient to compute large virtual rheological objects in real-time. This paper describes the real-time computation of the deformation of virtual rheological objects on an FPGA(Field Programmable Gate Array). An FPGA is an LSI in which logical circuit is rewritable. FPGAs enable us to perform computation of virtual rheological deformation in parallel. We designed a logical circuit to compute the deformation of virtual rheological objects on an FPGA, and we realized a system that computes the deformation 8.26 times as fast as a PC with 1.7 GHz CPU. Our estimations show that an FPGA is capable of computing the deformation 75.2 times faster than a PC.**

Keywords : **FPGA, parallel computing, deformation, virtual objects**

1　はじめに

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 以上の更新頻度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 よりも高い更新頻度での計算が必要であることが報告されている [1].

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，本論文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，著者らは，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．

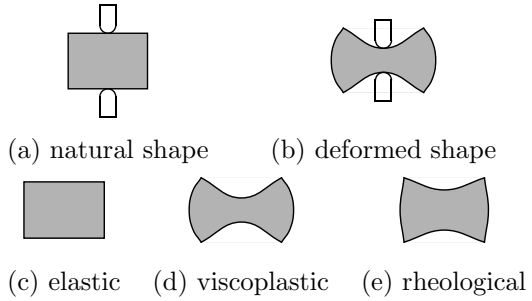


図 1　弾性物体，粘塑性物体，レオロジー物体
Fig. 1　Viscoelastic, plastic, and rheological objects

1.1　レオロジー物体の定義

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．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

[*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.

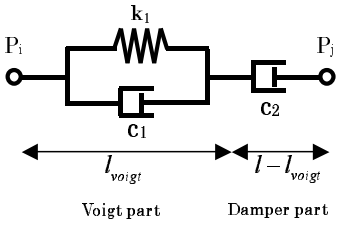
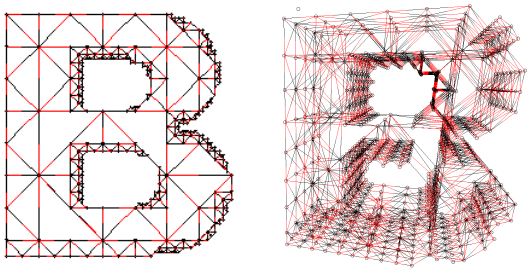


図 2　三要素モデル
Fig. 2　Three element model



(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

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-(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 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，本論文では，レオロジー物体と定義する．

1.2　先行研究

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 [2]．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 [3, 4]，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 [5]．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．

ハードウェアによる高速化の例として，近年では GPU が一般計算に使われ始めている [8, 9]．GPU は一般消費者向けに量産されるため比較的安価であり，GPU の利用はコンシューマを対象とした場合には優れた手法である．しかし，コンシューマを対象としない場合は，必ずしも現行の GPU の性能の枠に囚われる必要はない．

そこで，本論文では，大規模な物体の変形をリアルタイムに計算する手法として FPGA を用いる．

2　仮想レオロジー物体の力学モデル

本節では，レオロジー物体の力学モデルについて述べる．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．弾性物体の変形においては，初期の弾性理論の導入 [6, 7] 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 [10]．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．そこで本論文では，よりシンプルなモデルとしてバネ質点モデル [11] をベースにしたレオロジー物体の動力学モデル [12] を用いる．

レオロジー物体の性質を表現できる力学要素の内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 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．二次元物体および三次元物体は，図 3 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．

次に，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，

$$l_{\mathrm{damper}} = l - l_{\mathrm{voigt}} \tag{1}$$

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，

$$f_{\mathrm{voigt}} = -k_1(l_{\mathrm{voigt}} - L) - c_1\dot{l}_{\mathrm{voigt}} \tag{2}$$

$$f = -c_2 \dot{l}_{\mathrm{damper}} \tag{3}$$

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)，(3) 式より，フォークト部の長さの微分 $\dot{l}_{\mathrm{voigt}}$ 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，

$$\dot{l}_{\mathrm{voigt}} = A\left(l_{\mathrm{voigt}} - L\right) + B\dot{l} \tag{4}$$

となる．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 $\boldsymbol{e}_k$ 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 $f_k\boldsymbol{e}_k$ 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 $-f_k\boldsymbol{e}_k$ に一致する．したがって，質点 P_i の運動方程式は，

$$m_i\dot{\boldsymbol{v}}_i = \boldsymbol{f}_{\mathrm{R}i} + \boldsymbol{f}_{\mathrm{S}i} + \boldsymbol{F}_i^{\mathrm{ext}} \tag{5}$$

となる．ただし，m_i は質点 P_i の質量，$\boldsymbol{v}_i$ は質点 P_i の速度，

$$\boldsymbol{f}_{\mathrm{R}i} = \sum_{k\in R_i} f_k\boldsymbol{e}_k, \quad \boldsymbol{f}_{\mathrm{S}i} = -\sum_{k\in S_i} f_k\boldsymbol{e}_k$$

であり，$\boldsymbol{F}_i^{\mathrm{ext}}$ 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，(4)(5) 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．本論文では，数値解法としてオイラー法を用いる．

3　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．

3.1　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．本論文では，FPGA として Virtex-II XC2V6000 を用い，FPGA 搭載 PCI ボードとして，東京エレクトンデバイス株式会社製の PC-BD-PCI2DVI を用いる．ただし，今回の回路は PC-BD-PCI2DVI に依存するものではなく，Xilinx 社の FPGA を搭載する他の 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．

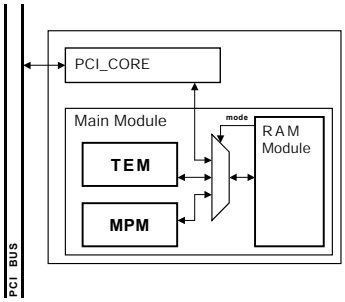


図 4　レオロジー物体変形計算回路概要
Fig. 4　Outline of circuit for rheoligical simulation

3.2　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 4 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，Xillinx 社が提供する既存の回路を用いる．

TEM は入力ポートから，第 k 稜線の始点 P_i の位置 $\boldsymbol{x}_i(t)$ と速度 $\boldsymbol{v}_i(t)$ ならびに終点 P_j の位置 $\boldsymbol{x}_j(t)$ と速度 $\boldsymbol{v}_j(t)$，稜線に対応する三要素モデルのフォークト部の長さ $l_k^{\mathrm{voigt}}(t)$ を取得し，出力ポートに $f_k(t)\boldsymbol{e}_k(t)$ と $l_k^{\mathrm{voigt}}(t+h)$ 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 $f_k(t)\boldsymbol{e}_k(t)$ は逐次，RAM module 内に蓄積され，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 $\boldsymbol{f}_i^{\mathrm{sum}}(t) = \boldsymbol{f}_{\mathrm{R}i}(t) + \boldsymbol{f}_{\mathrm{S}i}(t)$ 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 $\boldsymbol{f}_i^{\mathrm{sum}}(t) + \boldsymbol{F}_i^{\mathrm{ext}}$ および，質点 P_i の位置 $\boldsymbol{x}_i(t)$，速度 $\boldsymbol{v}_i(t)$ を取得し，出力ポートに $\boldsymbol{x}_i(t+h)$ と $\boldsymbol{v}_i(t+h)$ 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．

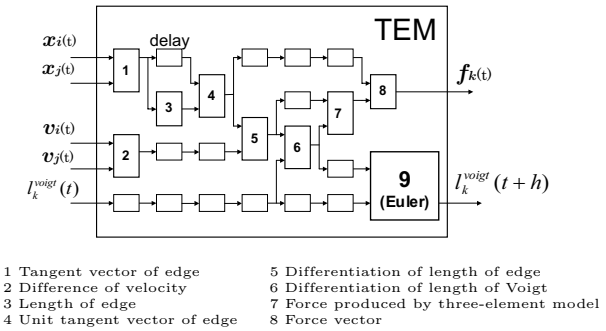


図 5　三要素モデル計算モジュール
Fig. 5　Module for computation of three-element models

3.3　TEM

TEM の詳細を図 5 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 $\boldsymbol{x}_k^{\mathrm{rel}}(t) = \boldsymbol{x}_j(t) - \boldsymbol{x}_i(t)$ を計算する．回路 2 では, 質点 P_i, P_j の相対速度 $\boldsymbol{v}_k^{\mathrm{rel}}(t) = \boldsymbol{v}_j(t) - \boldsymbol{v}_i(t)$ を計算する．回路 3 では, 三要素モデルの長さの逆数 $l_k^{\mathrm{inv}}(t)$ を求める．すなわち,

$$l_k^{\mathrm{inv}}(t) = \frac{1}{\sqrt{\boldsymbol{x}_k^{\mathrm{rel}}(t) \cdot \boldsymbol{x}_k^{\mathrm{rel}}(t)}} \tag{6}$$

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 [13]．回路 4 では三要素モデルの単位方向ベクトル

$$\boldsymbol{e}_k(t) = l_k^{\mathrm{inv}}(t)\boldsymbol{x}_k^{\mathrm{rel}}(t) \tag{7}$$

を求める．回路 5 では稜線の長さの微分 $\dot{l}_k(t) = \boldsymbol{e}_k(t) \cdot \boldsymbol{v}_k^{\mathrm{rel}}(t)$ を計算する．回路 6 では, (4) 式を用いて, フォークト部の長さの微分 $\dot{l}_k^{\mathrm{voigt}}(t)$ を計算し, 回路 7 では, (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 $\boldsymbol{F}_k(t) = f_k(t)\boldsymbol{e}_k(t)$ 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 $l_k^{\mathrm{voigt}}(t+h)$ を求める．すなわち,

$$l_k^{\mathrm{voigt}}(t+h) = l_k^{\mathrm{voigt}}(t) + h\dot{l}_k^{\mathrm{voigt}}(t) \tag{8}$$

を計算する．

3.4　MPM

MPM の詳細を図 6 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 $\dot{\boldsymbol{v}}_i$ および質点位置の微分 $\dot{\boldsymbol{x}}_i$ 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 $\dot{\boldsymbol{v}}_i(t)$ を,

$$\dot{\boldsymbol{v}}_i = m_{\mathrm{inv}}\boldsymbol{f}_i^{\mathrm{sum}} \tag{9}$$

で計算する．また，位置の微分は $\dot{\boldsymbol{x}}_i(t) = \boldsymbol{v}_i(t)$ で求める．

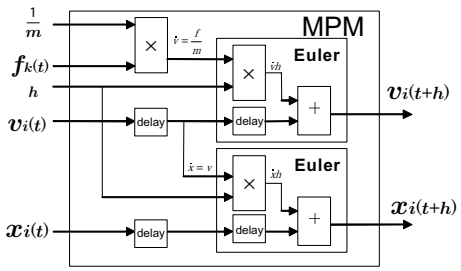


図 6　質点運動計算モジュール
Fig. 6　Mass particle module

3.5　RAM Module

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 $\boldsymbol{x}_i$, $\boldsymbol{v}_i$, $\boldsymbol{f}_{\mathrm{R}i}$, $\boldsymbol{f}_{\mathrm{S}i}$, $\boldsymbol{F}_k^{\mathrm{ext}}$, 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 $\boldsymbol{f}_{\mathrm{R}i}$ と $\boldsymbol{f}_{\mathrm{S}i}$ 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．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 $n_{\mathrm{mr}} = [(n_{\mathrm{m}} - 1)/1024] + 1$ であり, 質点に関するデータである $\boldsymbol{x}_i$, $\boldsymbol{v}_i$, $\boldsymbol{f}_{\mathrm{R}i}$, $\boldsymbol{f}_{\mathrm{S}i}$ および $\boldsymbol{F}_i^{ext}$ を格納するのに必要な BlockRAM の数はそれぞれ $3n_{\mathrm{mr}}$ となる．また, n_{e} 個の値の格納に必要な BlockRAM の個数は $n_{\mathrm{er}} = [(n_{\mathrm{e}} - 1)/1024] + 1$ 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 $2n_{\mathrm{er}}$ 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,

$$n_{\mathrm{br}} = 15n_{\mathrm{mr}} + 3n_{\mathrm{er}}. \tag{10}$$

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, $n_{\mathrm{mr}} = 1$ 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

表 1　単位変換表
Table 1　Unit conversion

	s	rs
Time:	$t[s]$	$t/r[rs]$
Elasticity:	$k[kg/s^2]$	$kr^2[kg/rs^2]$
Viscocity:	$c[kg/s]$	$cr[kg/rs]$
Force:	$f[kgm/s^2]$	$fr^2[kgm/rs^2]$
Velocity:	$v[m/s]$	$vr[m/rs]$

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．

4　誤差の抑制

今回の回路には, 固定小数点を用いるため, 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．例えば，更新頻度 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, 本論文で用いる 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．

4.1　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．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．そこで本論文では時間に対して, 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 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．

4.2　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, オイラー法の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．JIS 丸めは四捨五入丸めよりも高精度であるが, 本論文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．

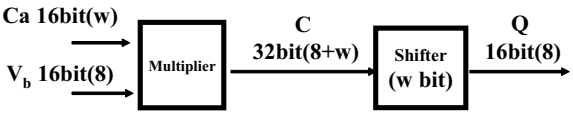


図 7　固定小数点位置を最適化した定数乗算回路
Fig. 7　Multiplier of constant value using optimized fixed decimal point

4.3　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 7 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．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,

$$w \leq 16 \tag{11}$$

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,

$$|c_\mathrm{a}| \leq 2^{15-w} - 2^{-w} \tag{12}$$

を満たす必要がある．よって, (11) 式および (12) 式から定数値における最適な小数点位置は,

$$w_\mathrm{mod} = \begin{cases} 14 - [log_2|c_\mathrm{a}|] & (14 - [log_2|c_\mathrm{a}|] \leq 16) \\ 16 & (14 - [log_2|c_\mathrm{a}|] > 16) \end{cases} \tag{13}$$

となる．これを演算に適用するには, 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．

通常の小数部 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 8 に示

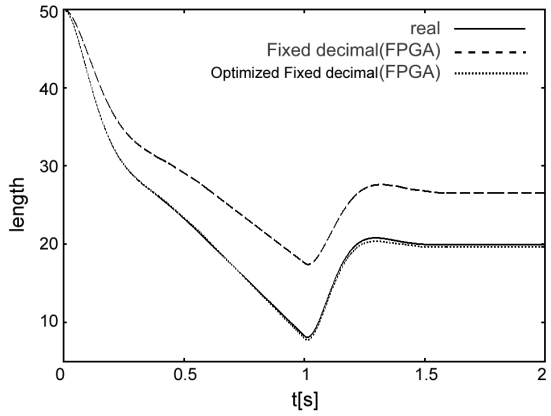


図 8　小数点位置最適化による三要素モデル変形推移の変化
Fig. 8　Deformation of three element model simulated with optimized fixed decimal point

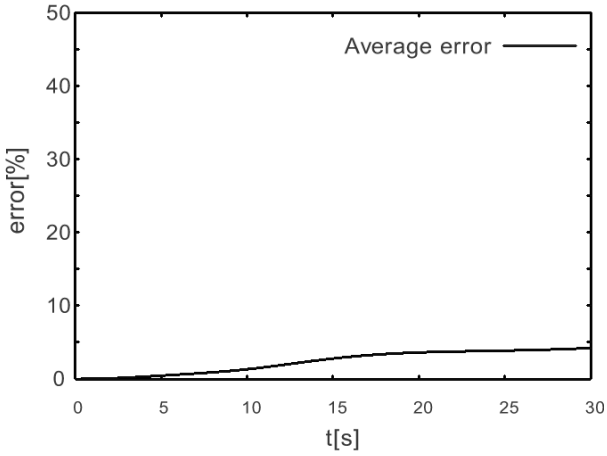


図 9　三次元物体変形過程における平均位置誤差
Fig. 9　Average positional error in simulation of 3D deformation

表 2　レオロジー物体変形計算回路の回路規模
Table 2　Size of circuits

	flip-flops	slices	multipliers
All modules	8,720	6,875	25
TEM	6,880	4,144	16
MPM	1,045	640	9

表 3　各軸質点数 9 × 9 × 9 の物体における FPGA と PC の処理時間の比較
Table 3　Computational time

	time[ms]
PC(Pentium4 1.7GHz)	1.14
FPGA(XC2V6000)	0.138

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の固定小数点では, 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．今回の手法は MPM の誤差の抑制には効果的であるが，TEM では定数に小さな値を用いないため顕著な効果を発揮しないと考えられる．そのため，今回は MPM のみに小数点位置の最適化を導入している．

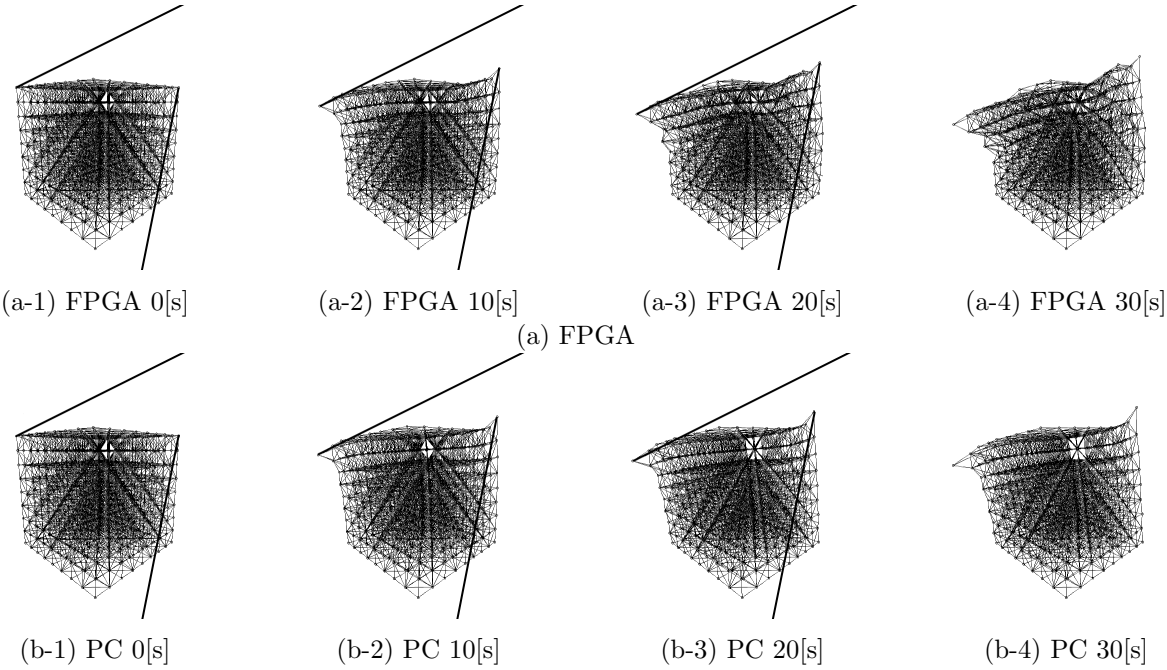
5　FPGA 実装

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 に示す．この回路は，XC2V6000 の回路の 20%を占める．

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．シミュレーションの対象として, 各軸 $9 \times 9 \times 9$ の質点を持つ立方体状の物体を用い，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 の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，$\boldsymbol{F}_0^{\mathrm{ext}} = [6.0, 30.0, 6.0]^T$ および $\boldsymbol{F}_1^{\mathrm{ext}} = [-30.0, -15.0, -30.0]^T$ を時刻 0 から時刻 30 までの間，物体上面の二点に加えた．

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 10 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 9 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 の計算時間の比較を行った．用いた PC は，Pentium4(1.7GHz) および 2GByte のメインメモリを搭載したものである．OS は Windows2000 であり，シミュレータは VisualC++6.0 の Release ビルドでコンパイルした．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 3 が得られた．FPGA 実機において，当該 PC の 8.26 倍の処理能力が実現されている．



(a-1) FPGA 0[s]　(a-2) FPGA 10[s]　(a-3) FPGA 20[s]　(a-4) FPGA 30[s]

(a) FPGA

(b-1) PC 0[s]　(b-2) PC 10[s]　(b-3) PC 20[s]　(b-4) PC 30[s]

(b) PC

図 10　仮想レオロジー物体の変形形状の比較
Fig. 10　Deformed shape of rheological objects

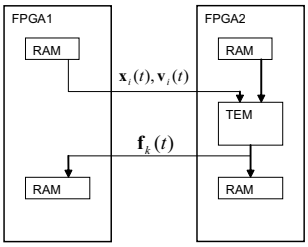


図 11　FPGA 間データ転送概略図
Fig. 11　Outline of data connection between a couple of FPGA's

6　実現可能な処理能力

本論文のような計算システムにおいては，シミュレーションに要する計算時間を，計算に要する実クロック数を元に明確に試算することができる．本節では，現行のシステムで実現可能な処理能力について述べる．

6.1　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, 片方の Main module において１クロック分のロスを生じる．ただし，グループ間にまたがる稜線の数は少なく，総計算時間に与える影響は大きくはない．

また，Main module 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．また各 FPGA に立方体状の物体を格納し，これを FPGA 間で接続する場合, 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．これは任意の数の FPGA で並列計算が可能であるということを意味する．

二つの FPGA の間で交換されるデータは図 11 で表されるように，$\boldsymbol{x}_i(t)$,$\boldsymbol{v}_i(t)$, $\boldsymbol{f}_k(t)$ の 9 変数各 16bit である．単純に計算すると，必要となる配線の数は 144 本であり，これが六面となると，864 本になる．ただし，データは必ずしも１クロックで送る必要はない．二グループにまたがるエッジの本数は全体から見ると少なく，一つの FPGA に１５×１５×１５の物体を格納した場合において，一つの面で 645 稜線である．全稜線の数は 19531 であり，境界データの比率は 3.3 %に過ぎないので，ワーストケースの場合でも 30.3 クロックに一回，データを転送できればよい．そのため，

必要な配線数は一組の FPGA の間で 5 本であり，6 つの FPGA と接続した場合でも 30 本に抑えられる．

6.2　現行のシステムにおける処理能力の試算

現行のシステムで実現可能な処理能力について述べる．本論文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,

$$c_{\mathrm{te}} = \frac{n_{\mathrm{es}} + 2n_{\mathrm{et}}}{P_{\mathrm{m}}} + d_{\mathrm{te}} \tag{14}$$

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る．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り，上式は，TEM の並列化によって $n_{\mathrm{et}}/P_{\mathrm{m}}$ クロックのロスが生じることを表している．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．TEM はレイテンシ 33 の CORDIC コアとレイテンシ 32 の除算回路および，乗算回路，加算回路その他で構成されており，TEM 全体のレイテンシは $d_{\mathrm{te}} = 90$ である．

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．

$$c_{\mathrm{mm}} = \frac{n_{\mathrm{m}}}{P_{\mathrm{m}}} + d_{\mathrm{mm}} \tag{15}$$

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．また，今回の MPM において d_{mm} は 8 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,

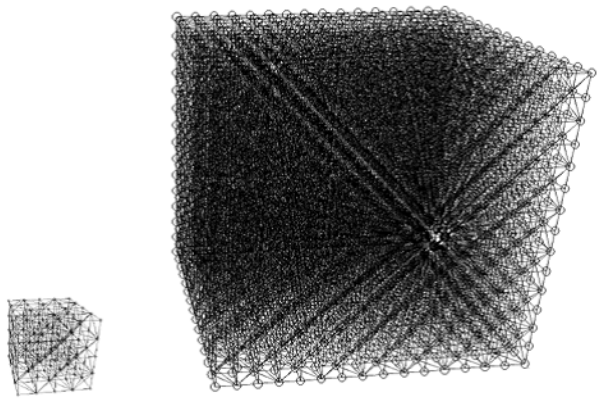
$$t_{\mathrm{s}} = \frac{c_{\mathrm{te}} + c_{\mathrm{mm}}}{r_{\mathrm{c}}} \tag{16}$$

となる．

本論文の回路では，$d_{te} = 90$，$d_{mm} = 8$，$r_c = 33MHz$ 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, 3.5 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．本論文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，更新頻度 1000Hz での計算が可能である．

3.5 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



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

図 12　更新頻度 10kHz で計算可能な最大規模の物体
Fig. 12　Maximum computable virtual object size in 10kHz

１セット分の BlockRAM を要するため, 並列可能な Main module 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，ISE5.1 を用いて，Main-Module を 66MHz の制約下の元で配置配線することが可能であることを確認している．100MHz および 133MHz では，制約を満たすことができず，現実的な動作周波数は 66MHz である．XC2V6000 に内蔵される DCM(Digital Clock Manager) を用いて MainModule を 66MHz で駆動させることが可能であると考えられる．

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，更新頻度 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 12 に載せる．図 12-(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 12-(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 Pentium4 1.7GHz 搭載 PC の 75.2 倍と試算される．

7　おわりに

大規模な仮想レオロジー物体の変形シミュレーションをリアルタイムで行うために，FPGA を用いた並列計算を提案した．仮想物体変形計算回路の設計と実装を行い, FPGA 実機で, Pentium4(1.7GHz) 搭載 PC

の 8.26 倍の計算速度での物体変形シミュレーションを実現した．また, 現行のシステムで, 同 PC の 75.2 倍の処理能力が得られることを試算した．

本論文において，任意の数の FPGA を並列化可能であることを述べた．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．

また, FEM による物体変形シミュレーションも FPGA による高速化の対象となりうる．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が, 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, 本論文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．

ただし，固定小数点では必ずしも計算精度が十分ではなく，メッシュの分割数が多い場合，稜線が短くなるため誤差が増大する．そのため，今後は FPGA においても浮動小数点を用いることが望ましい．浮動小数点の加算，減算，シフト等の処理は FPGA でも比較的小さな回路で扱うことができると考えられる．しかし，浮動小数点による乗算，除算および平方根演算を FPGA の基本ロジックで実現するには必要となるロジック数が多くなり，現状の FPGA には荷が重い．従って，FPGA で浮動小数点によるシミュレーションを行うための方策として，1)FPGA への浮動小数点演算器の内蔵，2)ASIC による浮動小数点パッケージとの接続，の二点が考えられる．

離散要素法およびパーティクルベースモデルのシミュレーションにおいては，必ず距離ベクトルと単位ベクトルの計算が必要となる．距離ベクトルの計算式は平方根を一つ含み，単位ベクトルの計算式には除算計算を一つ含むため，Main module 一つに対し，除算回路と CORDIC 回路各一つが必須である．従って FPGA に接続する ASIC には，距離ベクトルおよび単位ベクトルのベクトル演算回路か，除算回路および CORDIC 回路が搭載されていることが望まれる．必要な配線の本数や用途の広さから考えると，FPGA には後者が適すると考えられる．これらを用意することで，FPGA を用いて PC と同等の計算精度で高速なシミュレーションを実行することが可能になると考えられる．

謝辞

本研究は，科学研究費補助金（課題番号 14205039）の補助を受けた．

参考文献

[1] Katsuhito Akahane, Shoichi Hasegawa, Yasuharu Koike, 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).

[2] Makino Junichiro, Fukushige Toshiyuki, Koga Masaki, Namura Ken: GRAPE-6: The massively-parallel special-purpose computer for astrophysical particle simulation, PASJ, Vol.55, No.6(2003).

[3] D. L. James, D. K. Pai, *ArtDefo Accurate Real Time Deformable Objects*, Computer Graphics Proceedings (SIGGRAPH 1999), pp.65-72, (1999).

[4] D. L. James, D. K. Pai, Real Time Simulation of Multizone Elastokinematic Models, Proc. IEEE Int. Conf. Robotics and Automation, pp.927-32, (2002).

[5] G. Debunne, , M. Desbrun, M.-P. Cani, A. H. Barr, *Dynamic Real-Time Deformations using Space & Time Adaptive Sampling*, Computer Graphics Proceedings (SIGGRAPH 2001), pp.31-36, (2001).

[6] D. Terzopoulos, J. Platt, A. Barr, K. Fleischer: Elastically Deformable Models, Computer Graphics, Vol.21, No4, pp.205-214, (1987).

[7] D. Terzopoulos, A. Witkin, *Deformable Models Physically Based Models with Rigid and Deformable Components*, IEEE Computer Graphics and Applications, November, pp.41-1, (1988).

[8] M. de Pascale, G. de Pascale, D. Prattichizzo, F. Barbagli, *A GPU-friendly Method for Haptic and Graphic Rendering of Deformable Objects*, Proceedings of Eurohaptics 2004, 2004, pp. 44-51.

[9] A. Moravanzky, *Dense Matrix Algebra on the GPU*. ShaderX^2, 2003.

[10] D. Terzopoulos, K. Fleischer, *Modeling Inelastic Deformation: Viscoelasticity, Plasticity, Fracture*, Computer Graphics, Vol.22, No.4, pp.269-278, (1988).

[11] D. Terzopoulos, K. Waters, *Physically-based facial modeling, analysis, and animation*, Visualization and Computer Animation, vol.1, No.2, pp.73-80, (1990).

[12] 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).

[13] Jack E. Volder: *The CORDIC Trigonometric Computing Technique*, IRE Trans. Electronic Computers, Vol EC-8, pp.330-334, (1959).

(2005 年 2 月 9 日受付)

［著 者 紹 介］

友國　誠至

2005 年 立命館大学大学院博士前期課程卒業．在学中，レオロジー物体のモデリングおよび FPGA を用いた高速計算に関する研究に従事．修士 (工学)

平井　慎一　（正会員）

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 立命館大学理工学部 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)