

# 境界マップを用いた高速な流体弾性体連成シミュレーション

## Fast Coupling of Fluid and Deformable Bodies using Boundary Mapping

今村 紀之<sup>†</sup> 原田 隆宏<sup>‡</sup> 河口 洋一郎<sup>‡</sup>

Noriyuki IMAMURA<sup>†</sup>, Takahiro HARADA<sup>‡</sup> and Yoichiro KAWAGUCHI<sup>‡</sup>

<sup>†</sup> 東京大学 学際情報学府 <sup>‡</sup> Graduate School of Interdisciplinary Information Studies, the University of Tokyo

<sup>‡</sup> 東京大学 情報学環 <sup>‡</sup> Interfaculty Initiative in Information Studies, the University of Tokyo

E-mail: <sup>†</sup> cask.jp@gmail.com, <sup>‡</sup> {takahiroharada, yoichiro}@iii.u-tokyo.ac.jp

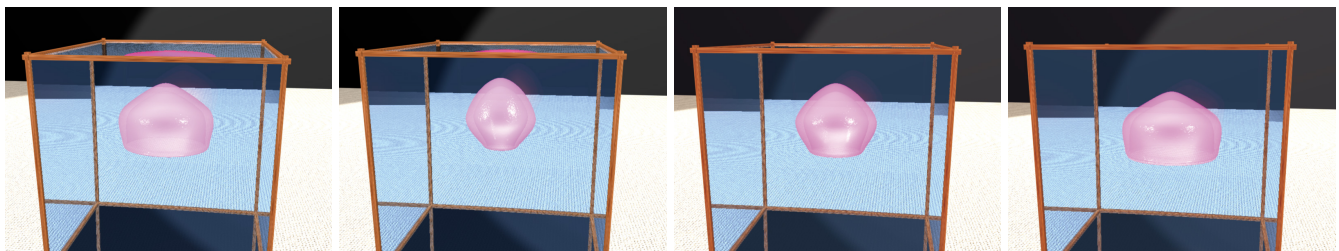


図1 流体中の弾性体のアニメーション：本手法を用いて2次元でシミュレーションした結果を疑似的に3次元化したもの

## 1 序論

### 1.1 背景

近年、CM や映画、ゲームなどで写実的なコンピュータ・グラフィックス (CG) 映像を用いた作品を目にする機会が増えてきた。これらの作品では、その見た目の確からしさのために物理シミュレーションが用いられていることもある。とりわけ、流体や弾性体のように自由に形を変えることのできる物体を直接手でモデリングし、尤もらしい動きを作成するのは至難の業であるが、物理シミュレーションを用いれば自然な動きを計算することができる。そのため、物理シミュレーションに対する需要は極めて高い。

流体や弾性体をそれぞれ計算する基礎的なシミュレーションする手法については数多く研究されている一方で、両者のインタラクションを扱った手法については未だ緒についたばかりである。

### 1.2 関連研究

CG における物理シミュレーションの応用に関して、Stam[13] が計算力学の分野から移流の解法にセミ・ラグランジュ法を導入し、構造格子を用いた流体シミュレーションを行った。Stam の手法は、CG における流体シミュレーションの基礎技術として様々な用途に応用が図られている。本研究においても、流体部分の手法は Stam の手法を基本として拡張を行っている。

一方、弾性体においても 1987 年には既に、Terzopoulos ら [15] によってシミュレーション手法が提案されている。また、O'Brien と Hodgins[12] は弾性体の脆性破壊のシミュレーション

および可視化の手法を提案した。同研究においては、陽的有限要素法 (explicit Finite Element Method: explicit FEM) を用いた。

これらの手法を背景に、流体と固体の連成に関しても近年盛んに研究が行われている。固体を動かない「壁」として扱うことで、固体から流体への影響は計算することができる。Mei ら [9] は、Lattice Boltzmann Method によるシミュレーションにおいて、複雑な形状をもった境界を扱う手法を開発した。

このような one-way の連成手法に対し、高橋ら [14] は Cubic Interpolated Propagation 法を用いた流体と固体の two-way のインタラクションを Volume of Fluid 法を用いて実現した。

また、Carlson ら [3] は、剛体を「剛体制約条件を化した流体」とみなすことで、剛体を流体のように解いた。このように、流体と固体を統一的なフレームワークでとらえ、モデル化する手法を用いると、境界部分における計算コストを最小限に抑えることができる反面、固体が軽い場合には安定に解くことができなかったり、流れの速い部分や固体によって栓をされているような部分では速度場や流体そのものが流出してしまうような現象が起こってしまうという欠点がある。

前述のいずれの研究においても連成対象となる固体は「剛体」であるが、Müller ら [11] は、FEM を用いた弾性体と Smoothed Particle Hydrodynamics (SPH) 法による流体とのインタラクションを実現した。Müller らの手法は四面体要素の上に粒子を配置し、それらを用いて流体との相互作用の計算を行ったため、計算精度は高くない。

また、Chentanez ら [4] は、流体と弾性体のインタラクションについて、空間離散化手法に制約されない陰的および半陰

的な解法を提案した．Chentanez らの手法は流体と弾性体の連成を正確にシミュレーションできるが，四面体要素を用いて，あらかじめ恣意的に弾性体に沿ってメッシュを切る必要がある．また，1 タイムステップの計算に 30～150 秒ほどかかっており，リアルタイムアプリケーションに応用することはできない．

### 1.3 目的

以上の背景を踏まえ，本研究では，インタラクティブな応用に耐えうる，高速で安定な流体・弾性体の連成シミュレーション手法を提案することを目的とする．

境界部では流体から弾性体に働く力を計算し，two-way の流体・弾性体連成を行う．

また，流体については高速に計算可能な一様構造格子での実装を行う一方で，弾性体の各面を格子に沿って階段状に近似することは行わずに，境界速度を内挿によって得ることによって，境界部の差分精度を高める．

さらに，高速な手法の開発に際して，特に計算コストが高くなりがちな流体・弾性体の境界部においては，境界マップを導入することで，境界部の判定を高速に行う．

## 2 流体領域と弾性体領域のモデル化

### 2.1 流体領域のモデル化

高速に計算可能な手法として，流体領域は一様な正方スタガード格子を利用した有限差分法を用いる．2 次元ではセル・ベース，3 次元ではボクセル・ベースとも呼ばれ，CG においては，最も確立されている手法の一つである [13, 5, 7]．

#### 2.1.1 支配方程式

流体の支配方程式として，非圧縮性流れのニュートン流体における Navier-Stokes 方程式 (1) と連続の式 (2) を用いる．

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

$$\nabla \cdot \mathbf{u} = 0 \quad (2)$$

このとき， $\mathbf{u}$ ， $\rho$ ， $p$ ， $\mathbf{f}$ ， $\nu$  はそれぞれ，時刻  $t$  における着目した点での速度ベクトル，密度，圧力，単位体積当たりの外力ベクトル，動粘性係数である．また，非圧縮性の Navier-Stokes 方程式を解く際には  $\mathbf{u}$ ， $p$  がそれぞれ未知数となるため，式 (2) を連立させて解を得る．

Helmholtz-Hodge 分解定理 [1] を用いると，あるベクトル場  $\mathbf{w}$  を質量保存のベクトル場  $\mathbf{u}$  とスカラー場の勾配  $\nabla p$  に分解することができる．流体のシミュレーションの場合は以下のように中間的な速度場  $\mathbf{w}$  を質量保存の速度場  $\mathbf{u}$  と圧力場の勾配  $\nabla p$  に分解することができるということになる．

$$\mathbf{u} = \mathbf{w} - \nabla p \quad (3)$$

この発散をとると，

$$\nabla \cdot \mathbf{w} = \nabla^2 p \quad (4)$$

が導かれる．これは一般に圧力ポアソン方程式と呼ばれる方程式であり，この式を解いて圧力を計算し，圧力項の計算を行う．

### 2.1.2 Navier-Stokes 方程式の離散化

離散化の際に，式 (1) の右辺第一項にあたる移流項の計算には，Semi-Lagrange 法を用いる．また，粘性拡散項および圧力ポアソン方程式は陰的に計算する．この際，連立方程式の解法には Jacobi 反復法を用いる．

なお，圧力振動を抑えるためベクトルの物理量をグリッドの面上に定義するスタガード・グリッドを用いる．

各タイムステップでの流体の計算手順は以下になる．

1. 移流項の計算：式 (1) 右辺第 1 項
2. 粘性拡散項の計算：式 (1) 右辺第 3 項
3. 外力項の計算：式 (1) 右辺第 4 項
4. 圧力ポアソン方程式の計算：式 (4)
5. 圧力項の計算：式 (3)

### 2.2 弾性体領域のモデル化

CG における弾性体表現は，動的な対象を扱うため，時間発展的な解法である必要がある．そこで，本研究ではいわゆる explicit FEM と呼ばれる手法を用いる．explicit FEM では FEM の「各節点にかかる力」を陽的に解く手法であり，この力を時間積分することにより，時刻  $t$  における変位が求まる．弾性体のモデルとして CG 分野で広く用いられている [2, 12, 10]．

#### 2.2.1 ひずみとひずみ速度

弾性体のひずみの定義として，大変形微小ひずみを仮定し以下で表わされる Green のひずみを用いる．Green のひずみは 3 次元では  $3 \times 3$  のテンソルで，その  $ij$  成分は式 (5) のように表わされる．

$$\epsilon_{ij} = \left( \frac{\partial \mathbf{x}}{\partial \mathbf{m}_i} \cdot \frac{\partial \mathbf{x}}{\partial \mathbf{m}_j} \right) - \delta_{ij} \quad (5)$$

ここで，変位後のワールド座標での位置  $\mathbf{x}$  は，物質座標で表わされた物体の初期座標  $\mathbf{m}$  の関数としてあらわされるものとする．なお， $\delta_{ij}$  は Kronecker のデルタである．

また，ひずみ速度  $\dot{\epsilon}$  については式 (5) の両辺を時間微分することにより得られる．

#### 2.2.2 構成式

弾性体のモデルとして，バネとダッシュポットが並列に並んだ Voigt モデルを用いる．このとき，応力  $\boldsymbol{\sigma}$  は，ひずみに影響する項  $\boldsymbol{\sigma}^{(\epsilon)}$  とひずみ速度に影響する項  $\boldsymbol{\sigma}^{(\dot{\epsilon})}$  の和であるとみなすことができる．両者はそれぞれ弾性ポテンシャルおよび減衰ポテンシャルを空間微分することで得られ，定数  $\lambda$ ， $\mu$ ， $\lambda'$ ， $\mu'$  を用いて以下のように成分表記される．

$$\sigma_{ij}^{(\epsilon)} = \lambda \text{tr}(\boldsymbol{\epsilon}) \delta_{ij} + 2\mu \epsilon_{ij} \quad (6)$$

$$\sigma_{ij}^{(\dot{\epsilon})} = \lambda \text{tr}(\dot{\boldsymbol{\epsilon}}) \delta_{ij} + 2\mu \dot{\epsilon}_{ij} \quad (7)$$

#### 2.2.3 構成式の離散化

弾性体領域は 2 次元の計算では，三角形メッシュによる空間離散化を行い，各三角形要素での物理量は重心座標系であらわされる．

各要素で求められた応力と外力の和は節点に配分され，陽的に時間積分することによって速度と変位が求まる．陽的な手法

であるため、硬い弾性体の計算を行うときは小さなタイムステップを刻まなければ安定に計算することができない。そこで、本手法では1流体ステップ中で複数回サブステップを刻み、全体の計算効率を下げずに弾性体を安定に解いた。

各タイムステップでの弾性体の計算手順は以下のようになる。

1. 変位勾配・変形勾配速度テンソルの計算
2. 弾性応力・粘性応力の計算：式 (6)・式 (7)
3. 内力・外力の計算
4. 速度・変位の計算

なお、境界部で流体から受ける力は、手順3で外力として足し合わせる。

### 3 境界のモデル化

CG分野において、剛体と流体等での連成手法では、図2左にあらわされるように、境界を流体グリッドに合わせて階段状に近似することが多い[6, 8]。しかし、これを弾性体に適用しようとする、流体のすべての計算点で弾性体の各要素との内外判定を行わなければならない、計算コストが極めて高い。そこで、本手法では図2右にあらわされるような、境界マップという概念を提案する。境界マップでは、流体計算点から境界までの位置のみを差分スキームに取り込むことで、高速で、より精度の高い連成シミュレーションを実現する。

#### 3.1 境界マップ

境界マップとは、境界部の処理に必要な情報を、流体の計算点をキーとして、連想配列に格納したものである。

境界マップを用いると以下の3つの利点がある。

- 境界の判定の計算を効率化できる
- 境界条件の適用の際に境界までの距離を差分スキームに取り入れることができる
- 境界に作用する力を算出する際に、境界上の物体速度及び適用する境界辺を境界マップを参照することで得られる

#### 3.2 境界マップの作成

境界マップに格納されている個々の要素が保持すべき情報は以下のとおりである。

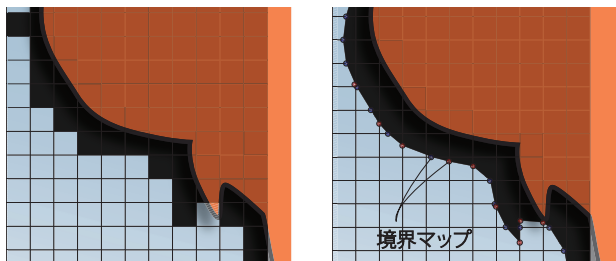


図2 従来の階段状に近似する手法(左)と境界マップを用いた手法(右)

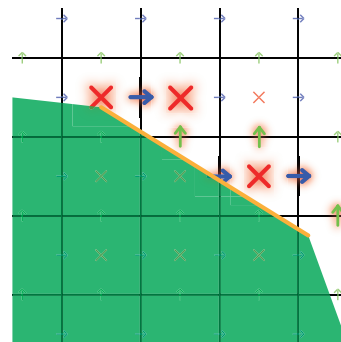


図3 物体境界と境界上の力の算出に用いる格子点

- 流体格子点から境界辺までの距離
- 流体格子点がどの境界辺の近傍としてみなされているのか
- 近傍としてみなされている境界辺の速度

このとき、流体の格子構造にはスタガード・グリッドを用いているが、実際に流体の速度および圧力を保持している点ではなく、横  $\Delta x/2$ 、縦  $\Delta y/2$  の間隔で区切られた格子における格子点すべてを「流体格子点」とみなす。

実際の境界マップの作成手順の概略を Algorithm1 に示す。

境界マップは各軸(2次元では  $x$  軸及び  $y$  軸)についてそれぞれ生成する。このとき、 $x$  軸及び  $y$  軸方向の境界マップは、それぞれ  $y$  成分および  $x$  成分をキーとしてそれぞれソートする。

境界マップは境界辺についてループして作成されるため、流体計算点について探索する場合に比べ計算コストが低い。

なお、境界は逐次移動するため、境界マップは各タイムステップで更新する必要がある。

#### 3.3 内部判定

通常、各流体計算点についてある軸方向にスキャンを行い、境界を跨いだかどうかで内外判定を行う。しかし、弾性体と流体とのインタラクションでは、弾性体が多数の要素からなっており、流体計算点はどの要素の境界点に当たるのかを判別する必要があるため、内部かどうかの判定には多大なコストがかかる。

しかし、本手法ではスキャンの際に、ソートされた境界マップを参照することで、それまでに通過したマップの要素が偶数であれば外部、奇数であれば内部と判定できる。さらに、この方法を拡張するとマップ内の奇数番目の要素と偶数番目の要素の間を内部として一度にビットを立てることで、より高速に内部判定を行うことができる。

#### 3.4 境界に作用する力

厳密には構造連成方程式を解いて境界に働く力を求める必要があるが、構造連成方程式は境界近傍の計算点の大きさの連立方程式を解く必要があり、インタラクティブな応用に際してはコストが高すぎる。

そこで、本研究では境界部に作用する力は、境界条件を自然に満たすような方向に働く力を境界領域において発生する力であると仮定し、定式化を行う[16]。

### Algorithm 1 境界マップの作成

```

procedure MAKEBOUNDARYMAP
  map  $yMap$  ▷  $y$  軸に沿ったマップ
  for all  $e$  in  $Edges$  do ▷ すべての境界辺  $e$  について
    for  $j \leftarrow \text{gridCoord}(e.a.y), \text{gridCoord}(e.b.y)+1$  do ▷ 端点  $a$  の  $y$  座標から端点  $b$  の  $y$  座標 (格子点での座標に換算) まで
       $i \leftarrow \text{getNearX}(e, j)$  ▷ 直線  $y = j$  と境界辺  $e$  の交点における弾性体外側の最近格子点の  $x$  座標を取得
       $yMap[\text{Point}(i, j)] \leftarrow \text{MapData}(e, \text{distance}, \text{velocity})$  ▷ 必要な情報を格子点をキーとしてマップに格納
    end for
  end for
  map  $xMap$  ▷  $x$  軸に沿ったマップ
  .....
end procedure

```

境界条件は、境界付近において、流体の速度と弾性体の速度が等しいという条件であり、これと整合性をとるために、境界マップを用いて、力の適用を行う。  $i$  軸 ( $i \in x, y$ ) 方向の境界マップ  $\mathcal{B}_i$  の要素  $s_i$  について、同じセルに属する流体の速度点  $P$  での速度  $\mathbf{u}_P$  をもちいて、境界での力  $\mathbf{f}$  は、

$$\mathbf{f} = \sum_{s_i} \alpha \frac{\mathbf{u}_P - \mathbf{u}_{s_i}}{\Delta t} \quad (8)$$

とモデル化される。ここで  $\alpha$  は、課す整合性の強さを表し、通常セルにおける物体の体積割合を用いる。これは、境界以外のすべて流体であるセルでは  $\alpha = 0$  で、 $\mathbf{f} = 0$  が成立するようにとった線形補間による力であるとみなすことができる。

弾性体はポリゴンで表わされているため、この力の適用をすべての境界セルと境界辺の間で行い、足し合わせることで各境界辺に流体から働く力を算出することができる。各境界セルとその近傍境界辺の探索に境界マップを用いることで  $O(N_E)$  の計算量で境界に働く力を算出できる。

例えば、図3のような場合では、太い矢印で示した速度点がオレンジの境界辺の近傍速度点とみなされ、これらの点に関して力の適用を行う。

なお、本研究においては、境界辺のスケールが流体の計算点の格子間隔に比べて非常に大きいという仮定をおき、すべてのセルについて  $\alpha = 0.5$  と近似した。

このとき、弾性体から流体への力の適用は、圧力ポアソン方程式を境界条件を用いて解くことにより自然に適用される。そのため、特に新たな力を導入する必要はない。

## 3.5 境界条件の適用

境界条件の適用の際には、境界マップを利用して、計算点間に位置する物体境界までの距離のみを差分スキームに取り込む手法を適用する。数値流体解析における手法 [17] をスタガード・グリッドにおける差分化に応用し、CG に適する形に簡略化して用いる。基本的には速度の境界条件は、境界上で法線方向の速度が等しいという no-slip (非すべり) 条件を用いている。

### 3.5.1 移流における境界条件

スタガード・グリッドにおける移流の計算では、ある計算点での速度は、周囲4点の速度から線形補間して求める。例えば  $x$  軸方向の速度をもつ面での  $y$  軸方向の速度は周囲の4点の  $y$

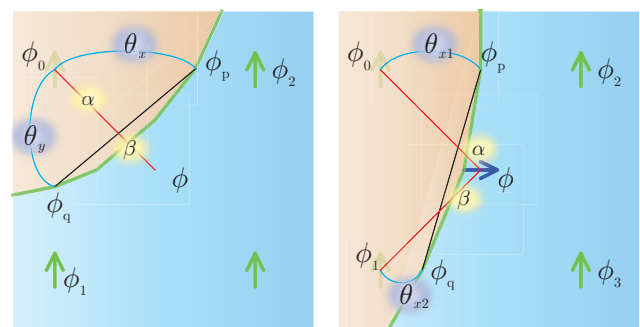


図4 移流計算に必要な周辺4点のうち1点(左)と2点(右)が利用できない場合

軸方向の速度から求める。

この4点のうち、1点(左)が利用できない、もしくは2点(右)が利用できない場合を考える。

図4のように、中央での物理量  $\phi$  を四方の物理量  $\phi_0, \phi_1, \phi_2, \phi_3$  を平均化して用いる場合、

$$4\phi = \phi_0 + \phi_1 + \phi_2 + \phi_3 \quad (9)$$

が成立する。1点  $\phi$  が利用できない場合(図4左)は、境界点での速度が流体と弾性体で等しいという条件を用いると、

$$\alpha\phi + \beta\phi_0 = \phi_q + \theta_x + \phi_p + \theta_y \quad (10)$$

が導かれる。ここで、 $\alpha = 2\theta_x\theta_y$ ,  $\beta = \theta_x + \theta_y - 2\alpha$  であり、 $\theta_x, \theta_y$  は  $\Delta x, \Delta y$  を1とした時の計算点と境界マップ点の距離である。式(9)および式(10)を連立して  $\phi_0$  を消去すると、以下の式を得る。

$$\phi = \frac{1}{\alpha + 4\beta} [\beta(\phi_1 + \phi_2 + \phi_3) + \phi_q\theta_x + \theta_y\phi_p] \quad (11)$$

2点  $\phi_0, \phi_1$  が利用できない場合(図4右)も同様に、

$$\phi = \frac{\phi_p + \phi_q + (1-k)(\phi_1 + \phi_3)}{2(2-k)} \quad (12)$$

を得る。ここで、 $k = \theta_{x1} + \theta_{x2}$  である。

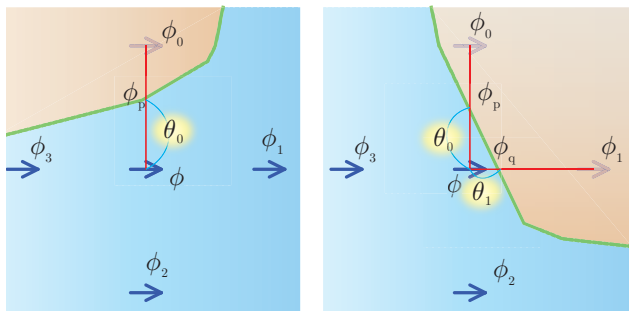


図5 勾配・発散計算に必要な周辺4点のうち1点が利用できない場合(左)と2点が利用できない場合(右)

表1 連成シミュレーションに用いた計算用のパラメータおよび計算時間

| 図  | サブタイム         |      | 格子解像度 |     | 弾性体要素数 | fps   |
|----|---------------|------|-------|-----|--------|-------|
|    | $\Delta t[s]$ | ステップ | 縦     | 横   |        |       |
| 図6 | 0.013         | 35   | 256   | 256 | 194    | 13.62 |
| 図7 | 0.05          | 100  | 256   | 256 | 160    | 9.0   |

### 3.5.2 粘性拡散と圧力ポアソン方程式における境界条件

粘性拡散の離散化および圧力ポアソン方程式においては、上下左右の4点を用いて計算を行う。このときに、1点 $\phi_0$ が利用できない場合(図5左)、4点の和 $\phi = \phi_0 + \phi_1 + \phi_2 + \phi_3$ は、

$$(1 - \theta_0)\phi + \theta_0\phi_0 = \phi_p \quad (13)$$

なる条件を用いて、

$$\phi = \phi_p + \theta_0(\phi_1 + \phi_2 + \phi_3) \quad (14)$$

を得る。同様に2点 $\phi_0, \phi_1$ が利用できない場合(図5右)は、境界上の物理量 $\phi_q$ を用いて、

$$\phi = \frac{\theta_0\theta_1(\phi_2 + \phi_3) + \theta_0\phi_q + \theta_1\phi_p}{\theta_0 + \theta_1 - \theta_0\theta_1} \quad (15)$$

となる。なお、形状が複雑になり3点が利用できなくなった場合も同様に導くことができる。

## 4 結果と考察

本手法を用いて、矩形状の棒の上端を固定し、下端を流体の噴出にさらす2次元のシミュレーションを行った。速度場と弾性体の挙動を可視化した結果を図6に示す。流体の噴出により棒が変形し、流体の速度場に影響を与える様子が見て取れる。

また、CGへの応用例として、2次元でのシミュレーション結果であっても、押し出しを使用することで図7のような疑似3次元のアニメーションを作成することができる。

なお、これらの計算は全てIntel Core 2 Duo 3.0GHzのCPUを搭載した計算機にて行なった。また、シミュレーションに用いたパラメータは表1に示したとおりである。

本手法では、境界をセルから境界までの距離のオーダーで近似しているため、境界部分での精度が向上し、速度の漏出現象は起こっていない。また、境界を階段状に近似する手法でおこ

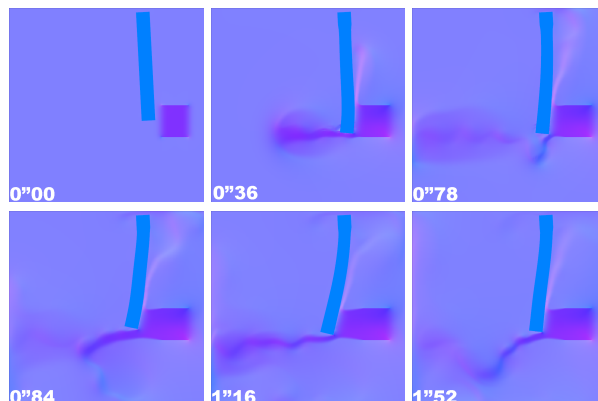


図6 弾性を有する棒と流れ場のインタラクションのシミュレーション結果。速度場を色で表示したもの。左下は実時間。

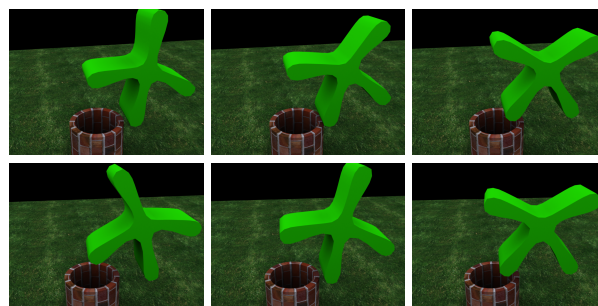


図7 土管から流出する流体によって回転する弾性体の風車のアニメーション：2次元でのシミュレーション結果を疑似的に3次元化し、3ds Maxを用いてレンダリングを行ったもの。

る、境界面の微小な段差による流れの阻害も、境界マップを用いることにより解消されている。

## 5 結論

流体と弾性体相互の影響を考慮に入れ、境界部で作用する力をモデル化するシミュレーション手法を開発した。本手法において、新たに境界マップを提案し、これを導入することにより、内部判定や境界条件の適用、境界で作用する力の計算が効率化され、流体と弾性体の連成を高速にシミュレーションすることができた。

本研究では、2次元でのシミュレーションを扱ったが、これを3次元へ拡張することで、より応用範囲が広がるものと考えられる。

CGへの応用に際しては、自由液面の表現は欠かせない。流体部分はスタガード・グリッドを用いた構造格子を使用しているため、Level Set法[5]と組み合わせることも可能である。

本研究では、境界を階段状近似せずに計算を行なった。しかしこの手法の計算精度や計算速度および安定性における定量的な比較評価が行われていないため、これらも今後の課題として挙げられる。

## 謝辞

本研究は科学技術振興事業団戦略的基礎研究推進事業 (CREST) の支援のもとに推進されました。

## 参考文献

- [1] G. Arfken. Helmholtz's theorem. In *Mathematical Methods for Physicists*, 3rd ed, pp. 78–84. FL: Academic Press, 1985.
- [2] D. Baraff and A. Witkin. Large steps in cloth simulation. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pp. 43–54, New York, NY, USA, 1998. ACM.
- [3] M. Carlson, P. J. Mucha, and G. Turk. Rigid fluid: animating the interplay between rigid bodies and fluid. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers*, pp. 377–384, New York, NY, USA, 2004. ACM.
- [4] N. Chentanez, T. G. Goktekin, B. E. Feldman, and J. F. O'Brien. Simultaneous coupling of fluids and deformable bodies. In *SCA '06: Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pp. 83–89, Aire-la-Ville, Switzerland, Switzerland, 2006. Eurographics Association.
- [5] R. Fedkiw, J. Stam, and H. W. Jensen. Visual simulation of smoke. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pp. 15–22, New York, NY, USA, 2001. ACM.
- [6] N. Foster and D. Metaxas. Realistic animation of liquids. *Graphical models and image processing: GMIP*, 58(5):471–483, 1996.
- [7] M. Harris. *Fast Fluid Dynamics Simulation on the GPU*, chapter 38. nVIDIA, 2004.
- [8] Y. Liu, X. Liu, and E. Wu. Real-time 3d fluid simulation on gpu with complex obstacles. *Computer Graphics and Applications, 2004. PG 2004. Proceedings. 12th Pacific Conference on*, pp. 247–256, 6–8 Oct. 2004.
- [9] R. Mei, W. Shyy, D. Yu, and L.-S. Luo. Lattice boltzmann method for 3-d flows with curved boundary. *Journal of Computational Physics*, 161(2):680–699, July 2000.
- [10] M. Müller, J. Dorsey, L. McMillan, R. Jagnow, and B. Cutler. Stable real-time deformations. In *SCA '02: Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pp. 49–54, New York, NY, USA, 2002. ACM.
- [11] M. Müller, S. Schirm, M. Teschner, B. Heidelberger, and M. Gross. Interaction of fluids with deformable solids. *Computer Animation and Virtual Worlds*, 15(34):159–171, 2004.
- [12] J. F. O'Brien and J. K. Hodgins. Graphical modeling and animation of brittle fracture. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pp. 137–146, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.
- [13] J. Stam. Stable fluids. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pp. 121–128, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.
- [14] T. Takahashi, H. Fujii, A. Kunimatsu, K. Hiwada, T. Saito, K. Tanaka, and H. Ueki. Realistic animation of fluid with splash and foam. *Computer Graphics Forum*, 22(3):391–400, 2003.
- [15] D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer. Elastically deformable models. In *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pp. 205–214, New York, NY, USA, 1987. ACM.
- [16] 梶島. 固体粒子が乱流に及ぼす影響の直接数値解析 (乱流構造の数値：発生・動力学・統計・応用). 数理解析研究所講究録, 1226:121–130, 2001.
- [17] 市川. 境界位置を取り込む直交格子差分法による 3 次元移動境界問題の計算. 第 15 回数値流体力学シンポジウム, 2001.