

特別研究報告書

インタラクティブ流体シミュレーション
における力覚提示の実装

指導教員 富田 眞治 教授

京都大学工学部情報学科

山口 明徳

平成19年2月9日

インタラクティブ流体シミュレーション における力覚提示の実装

山口 明德

内容梗概

近年の計算機性能の急速な向上に伴い、インタラクティブな実時間シミュレーションへの期待が高まっている。ここでは、オペレータによるシミュレーション対象へのインタラクティブな操作に対応して実時間でシミュレーションを行うとともに、即刻その結果を視覚その他の手段により提示することが求められる。流体シミュレーションは、工学、医学等の分野で重要視されており、研究が進められているが、計算量が多く、オペレータによる操作に動的に対応することが難しいため、流体シミュレーションをインタラクティブに実行しようとする試みはほとんど例がない。インタラクティブな流体シミュレーションを実現するためには、数値計算モデルの改良と計算の高速化により、オペレータの動的な操作に対して実時間で計算結果を提示しなくてはならない。

本稿では、まず、格子ボルツマン法を用いた流体計算処理と可視化処理を GPU 上で、抵抗力計算と力覚提示デバイスの制御を CPU 上でそれぞれ行うことによって、物体にかかる抵抗力を力覚提示デバイスに提示するとともに、実時間で流体中の物体の位置を変動させることのできる流体シミュレータを実装した。流体の運動にともなう数値計算を GPU 上で行うことによって、CPU による計算よりもはるかに高速な計算を実現することができた。この初期実装で力の計算には運動量交換法を用いた。しかし、この初期実装では物体の移動を停止させた場合に力が感じられないなど必ずしも流体力学的に正しい力覚の提示を行うことができないことが確認できた。

一般に、流体力学的抗力はいくつかの要因からなる複雑な現象である。コンピュータシミュレーションにおいて、これらの流体の流れに関する問題を本格的に解析するとなると、実時間でのシミュレーションが困難になってくる。Cg系の物理シミュレーションなどに用いられている流体の流れを巨視的に見た場合の近似計算を行うことによって、物体にかかる抗力の計算を行うことにした。これによって、流体と物体の相対速度の2乗に比例するという流体力学的な関係を満たす抗力を求めることができた。しかし、物体形状を変えると形状

によっては正しい抗力が求められない場合があるということが確認できた。また、物体形状による抗力の違いまでは正確に表現することができなかった。そこで、より高精度な圧力ベースの計算法の導入を検討した。

流れの中に置かれた物体、あるいは静止した流体中を運動する物体は流体から力を受ける。この力は、物体表面に働く圧力に基づく分布が物体の前後あるいは上下面で非対称なために生じる。したがって、物体周辺の圧力を積分することによって、物体にかかる抗力を求めることにした。格子ボルツマン法において、それぞれの格子点上の圧力は、密度 ρ に比例する。格子ボルツマン法の計算の 1 ステップごとに毎回それぞれの格子点での密度が求まっているので、圧力は簡単に求めることができ、実時間シミュレーションに対応可能な力覚の計算を行うことができる。しかし、物体周辺の密度分布を調べてみたところ、それぞれの格子点での密度は初期密度 ρ_0 がごくわずかしか変化しておらず、単純に圧力積分を行っただけでは、十分な物体への抗力が得られなかった。

ここで、圧力 P を Batchelor らが水の表現に用いた状態方程式を用いて、密度が平均密度以下の部分の圧力は小さく、平均密度以上の部分の圧力を大きくする補正を行うことによって、物体にかかる抗力を圧力により求める。この圧力ベースを用いた計算によって、流体と物体の相対速度の 2 乗、物体の基準面積に比例するという流体力学の性質を満たす抗力を求めることが可能となった。この補正により、計算量が若干増えるが、算術的な式変形を行うことによって、計算量の増大を最小限に抑えた。巨視的モデルに基づいた近似計算では表現できなかった物体形状による抗力の違いが表現できるようになった。さらに圧力ベースの計算法によって、物体への揚力も表現できていることが確認できた。

この圧力ベースの計算モデルをインタラクティブ流体シミュレータに用いることによって、より高精度な力覚の提示が可能なシミュレータを実装することができた。

Implementation of Force Sense Presentation for Interactive Fluid Simulation

Akinori YAMAGUCHI

Abstract

Recently, interactive simulation is coming to be done in the various fields, caused by the rapid improvement of the computer performance. This system needs to perform a simulation in realtime corresponding to the interactive operation by the operator and to show the result at once by means of vision and others needs. Fluid simulation is regarded as important in many fields such as engineering or medicine, nevertheless fluid simulation is rarely applied for interactive simulation, because of its high computational cost and difficulty of coping with interactive operation. To be applied for fluid simulation to interactive simulation, it is necessary to use suitable numerical model and to accelerate computing.

First, we implemented interactive fluid simulation by using GPU for calculation of fluid by lattice Boltzmann method and for process to be visible, and by using CPU for calculation of force and for control of a device for the sense of touch. This simulator shows force by a device for the sense of touch and we can move a object in fluid in real time. It could actualize much higher speed calculation by calculating about fluid on GPU than on CPU. It calculates the force by momentum exchange method. But, force calculating by momentum exchange method is not always correct in fluid mechanics.

In general, calculation of force to receive from fluid is very complex. This complex calculation makes simulation in realtime impossible. Therefore we calculate the force by Comprehensive fluid approximate calculation method. Force calculation by using this method is proportional to square of relative speed. This relation is correct in fluid mechanics. But according to the shape of the object, force isn't correct. And this method cannot express the different of force according to the shape of the object. Calculation method of the force due to pressure is introduced. This method require higher accuracy than approximate method.

Object in fluid is received force. It causes this force with pressure difference around object. Force is calculated by integrating pressure around object.

In lattice Boltzmann method, pressure in each points is proportional to density. We inspect density distribution around the object. Pressure in each points is nearly equal to mean density. Therefore, force calculated by integrating pressure around object is not force which it can be satisfied.

The force which depends on the object is calculated with pressure which is revised using the equation of state which Batchelor and others uses for expression of water. This revising pressure make pressure of the part below mean density smaller and pressure of part above mean density larger. It became possible to calculate the drag which satisfies the character of the aeromechanics that it is proportionate to the standard area of the object and square of relative speed of the fluid and the object with the calculation which uses this pressure base. With this revision, the computational complexity increases somewhat. But increase of the computational complexity was held down to minimum by doing arithmetic type deformation. This calculation method based on pressure can express the difference of the force with the object form which cannot be expressed in the approximate calculation based on the macroscopic model. Furthermore we verified that calculation by this pressure model can express the lifting force of object.

Using force calculation by pressure in interactive fluid simulation makes implementation of simulator which can be present high accurate force possible.

インタラクティブ流体シミュレーション における力覚提示の実装

目次

第1章	はじめに	1
第2章	研究の背景	2
2.1	流体運動の数値シミュレーション	3
2.1.1	格子ボルツマン法	3
2.2	格子ボルツマン法を用いた流体シミュレータ	5
2.2.1	格子と粒子速度モデル	6
2.2.2	固体壁における境界条件	7
2.2.3	GPUによる計算	9
2.2.4	触覚デバイス	9
2.3	流体中の物体にかかる抵抗	10
2.3.1	運動量交換法による力覚提示	10
2.3.2	運動量交換法での問題点	11
第3章	巨視的流体モデルに基づく力覚提示	13
3.1	相対速度と有効断面積	13
3.1.1	実行結果と考察	14
3.2	物体形状の変更	15
3.2.1	形状抵抗	15
3.3	相対速度	16
第4章	物体形状による抗力の違いをを反映した力覚の提示	19
4.1	圧力モデル	20
4.1.1	物体周辺の圧力に基づく抗力計算	20
4.1.2	実行結果と考察	21
4.1.3	揚力	22
4.2	インタラクティブシミュレーション	23
第5章	まとめ	24
	謝辞	25

第1章 はじめに

近年の計算機性能の急速な向上に伴い、インタラクティブな実時間数値シミュレーション [1] への期待が高まっている。中でも実時間内での数値シミュレーションは、大規模な3次元データを必要とする医療などの分野において高度な技術が要求されている分野である。このような中、次世代のシミュレーション技術として、従来の実験の代替手段となりえる「仮想実験型/仮想体験型のシミュレーション環境」の構築が望まれている。ここでは、オペレータによるシミュレーション対象へのインタラクティブな操作に対応して実時間でシミュレーションを行うとともに、即刻その結果を視覚その他の手段により提示することが求められる。

我々の身の回りには水や空気をはじめとした流体が多く存在し、特に工学や医学等の分野で流体の振舞いを知ることは重要な課題となっている。流体の振舞いに関しては、ごく単純な場合を除き、解析的に解を求めることは不可能であるため、計算機を用いた数値シミュレーションが利用される。しかし、流体シミュレーションをインタラクティブに実行しようとする試みは、ほとんど例がない。この要因として、流体計算は計算量が多く、実時間での実行が困難であることが挙げられる。また、格子の生成などにおいて、境界形状などに応じた問題の定式化が困難であり、オペレータの動的な操作に対応することが難しい。

インタラクティブな流体シミュレーションを実現するためには、オペレータの操作に動的に対応しうる数値計算モデルを使用することが必要となる。格子ボルツマン法 (Lattice Boltzmann Method)[2][3][4] は、数値流体力学の比較的新しい手法であり、局所的で単純な規則に従って計算が進む手法であることから、インタラクティブな操作に対応することが比較的容易であると考えられる。

また、実時間で計算結果を提示するためには、計算を高速化する必要がある。近年、汎用グラフィクスカードに搭載されるグラフィクスプロセッサユニット (Graphics Processing Unit, GPU) の性能は著しく向上しており、浮動小数点演算を正式にサポートするとともに、従来固定であったグラフィクス演算処理フローのプログラム制御までも可能となっている [5][6]。このため、GPU の本来の用途であるグラフィクス用途にとどまらず、GPU を汎用の数値計算にも応用しようとする試みが始められている [7]。

本研究では、まず、流体の運動を計算するモデルとして格子ボルツマン法、高

速計算のために GPU による計算，力覚の提示として触覚デバイスを用いたインタラクティブ流体シミュレーションを作成した．そして，力覚の提示部分の問題点をあげ，その問題を解決する手法として巨視的な流体モデルに基づいた力の計算によって力覚提示を行った．また，物体形状による問題，その解決策として圧力による力覚の提示の実装，考察を行い，最終的には正しい力覚を提示することが可能なインタラクティブ流体シミュレーションの実現を目指す．

第2章では，研究の背景として，格子ボルツマン法とそれを用いた流体シミュレーションと，その問題点について述べる．第3章では，巨視的流体モデルに基づく力覚提示と物体形状による問題点，第4章では，物体形状を反映した力覚の補正について述べ，最後に第5章でまとめる．

第2章 研究の背景

インタラクティブシミュレーションの適用例として，医療分野における手術シミュレータがある．医療の高度化，多様化が進む中，新たな教育法の確立が求められるようになってきた．このような中，現実の手術を模擬した仮想環境を用いて手術に必要な知識や技術の習得を効率的かつ安全に行う目的で手術シミュレータの実現に向けた研究が進められている．

人体は体重に対して約92%の体液などの流体で占められており，中でも血液の流れについては，日本での三大死因である心疾患や脳血管障害を引き起こす要因となりうることから，長年研究が進められてきた．近年では，このような人体における流体现象に数値シミュレーションを取り入れる研究が進められている．

インタラクティブな手術シミュレータに流体シミュレーションを取り入れることができれば，シミュレータによってより現実に近い仮想体験が得られ，教育目的にとどまらず，困難な技術や新しい技術を伴う手術におけるリハーサルが可能となり，術前計画にも役立たせることができる．しかし，流体シミュレーションをインタラクティブに実行することは一般に困難であるとされている．

本章では，本研究でインタラクティブな流体シミュレーションを実現するために流体の運動の計算モデルとして採用した格子ボルツマン法の計算手法，その実装について説明する．また，格子ボルツマン法による力の計算手法として研究されている運動量交換法を用いて力の計算を行った場合のシミュレーショ

ンについて述べる．

2.1 流体運動の数値シミュレーション

一般に流体の運動は，質量，運動量，エネルギーの保存に関するニュートン力学の基本法則に基づいた，Navier-Stokes 方程式によって定式化される．Navier-Stokes 方程式は一般に非線形の偏微分方程式となり，一方向の流れ，あるいは環状の流れなど方程式が線形になるものを除いて，厳密解はほとんど知られていない．したがって，差分法，有限要素法，境界要素法などの手法によって，Navier-Stokes 方程式を単純化した数理モデルを解くことになる．モデル化の過程で物体周りの空間を格子によって離散化するが，境界形状が複雑になれば，適切な格子の生成は非常に困難である．非構造格子（三次元では通常 4 面体セルの集合を用いる）を使うことで精度や速度が犠牲になるが，格子生成の自由度が高まり，ある程度格子の自動生成は可能になる．しかし，本研究で扱うインタラクティブシミュレーションでは，シミュレーション実行中のオペレータによる操作により，境界位置を動的に変更できることを前提としている．非構造格子を用いたとしても，時間刻み毎に境界位置の変化に応じた格子の再生成が必要となり，実行速度が大きく低下する．

このような Navier-Stokes 方程式を近似によって解く従来のアプローチに対して，比較的新しい手法である格子ボルツマン法が注目されつつある [4]．格子ボルツマン法では，流体の振舞いを局所的で単純な規則による仮想粒子の運動として捉えることで，Navier-Stokes 方程式を解いた場合と同等な結果を得ることができる．そのため，巨視的視点からの格子生成を含めた定式化が困難である場合にも適用できる可能性がある．

2.1.1 格子ボルツマン法

格子ボルツマン法においては，原則として空間は規則的な格子によって一様に離散化される¹⁾．時間についても離散化し，時間刻み Δt ずつ時間が進む．流体を，格子上に存在する仮想粒子の集合と捉える．粒子は格子点に存在し，衝突と呼ばれる過程によって一部の粒子は速度の方向を変え， Δt の間に静止，あるいは別の格子点へ移動する（これを並進と呼ぶ）．粒子は個別に捉えるのではなく，時刻 t ，位置ベクトル r の格子点上において粒子速度ベクトル e_i を持つ粒

¹⁾ 最近では不規則格子への適用も研究されている [8]．

子数密度分布を，実数値を持つ分布関数 $f_i(r, t)$ で表す．一連の粒子運動は，

$$f_i(r + e_i \Delta t, t + \Delta t) = f_i(r, t) + \Omega_i(f(r, t)) \quad (1)$$

と表され，これを格子ボルツマン方程式と呼ぶ．式 (1) の左辺は，衝突終了後に粒子が $e_i \Delta t$ 離れた近接の格子点に移動する並進過程を表している． Ω_i は，衝突による粒子分布の変化を表す衝突演算子である． Ω_i は，局所的な分布関数にのみ依存している．

格子点における密度 ρ と運動量 ρu は，粒子速度モーメントとして分布関数 f_i を用いて次のように定義される．

$$\rho = \sum_i f_i, \quad \rho u = \sum_i f_i e_i \quad (2)$$

ここで u は，格子点における巨視的流速である．これら巨視的変数は，式 (1) による衝突・並進過程の後，毎回算出される．

Ω_i は，それぞれの格子点において，質量と運動量を保存するように決定されなければならない．すなわち，

$$\sum_i \Omega_i(f(r, t)) = 0, \quad \sum_i \Omega_i(f(r, t)) e_i = 0 \quad (3)$$

Ω_i に関して，さまざまなモデルが提案されているが，単一緩和時間近似を用いた格子 BGK モデルと呼ばれるモデルが広く使用されている．このモデルでは，格子ボルツマン方程式は単一時間緩和係数 τ を用いて，

$$f_i(r + e_i \Delta t, t + \Delta t) = f_i(r, t) - \frac{f_i(r, t) - f_i^{eq}(r, t)}{\tau} \quad (4)$$

と表される．動粘性係数 ν は τ と音速¹⁾ c_s を用いて，

$$\nu = \left(\tau - \frac{1}{2} \right) c_s^2 \Delta t \quad (5)$$

と与えられる． $\nu > 0$ であることから， $\tau > 1/2$ でなければならない．格子ボルツマン法では完全な非粘性の流体を扱うことはできず， τ が $1/2$ に近づくと，数値安定性が失われる． f_i^{eq} は，局所平衡分布関数と呼ばれ，次に示す Maxwell-Boltzmann 平衡分布関数から導くことができる．

$$g^{eq} = \frac{\rho}{(2\pi RT)^{\frac{D}{2}}} \exp \left[-\frac{(\xi - u)^2}{2RT} \right] \quad (6)$$

¹⁾ ここでの音速とは密度変動の伝播速度であり，熱力学的な変数としての音速とは異なる．

D は空間の次元, ξ は粒子速度, R は気体定数, T は温度である. 巨視的流体変数は, 分布関数 g の速度モーメントによって与えられる. すなわち,

$$\rho = \int g d\xi, \quad \rho u = \int \xi g d\xi \quad (7)$$

ここで流速 u の大きさが c_s に比べて十分に小さい (すなわち Mach 数が十分小さい) と仮定し, 式 (6) を $O(u^2)$ まで展開すると,

$$g^{eq} = \frac{\rho}{(2\pi RT)^{\frac{D}{2}}} \exp\left(-\frac{\xi^2}{2RT}\right) \left[1 + \frac{\xi \cdot u}{RT} + \frac{(\xi \cdot u)^2}{2(RT)^2} - \frac{u^2}{2RT}\right] \quad (8)$$

音速を $c_s = c/\sqrt{3}$ として与え, 粒子速度を e_i に限定することによって離散化すれば, 式 (8) は,

$$f_i^{eq} = w_i \rho \left[1 + \frac{3(e_i \cdot u)}{c^2} + \frac{9(e_i \cdot u)^2}{2c^4} - \frac{3u^2}{2c^2}\right] \quad (9)$$

と書き換えられる. ただし, w_i は重み関数であり, 粒子の速さによって決まる.

$$w_i = \frac{1}{(2\pi c^2/3)^{\frac{D}{2}}} \exp\left(-\frac{3e_i^2}{2c^2}\right) \quad (10)$$

e_i, w_i は, Navier-Stokes 方程式に適合するように定められる. ここで用いた $\|u\| \ll c_s$ の仮定は, インタラクティブシミュレーションを実現する上での制限として注意しなければならない.

格子ボルツマン法の計算手法は, 便宜上, 衝突, 並進, 巨視化の 3 つの過程の繰り返しであると捉えることができる. このうち他の格子点との情報のやりとりは並進過程においてのみであり, 残りの過程の計算は全て格子点内で行われる. 格子 BGK モデルにおいては, 格子点内の他の速度方向を表す分布関数との干渉すら考慮する必要がない. 式 (1) もしくは式 (4) の左辺は並進を表すが, e_i の種類は極めて限られており, 使用する速度分布モデルによるが, 多くは $e_i \Delta t$ が隣接格子点までの距離のみを指す. このように, 格子ボルツマン法は局所的な計算手法であるため, 複雑境界の形状に対しても対応が容易であり, 並列計算にも向いているといわれている.

2.2 格子ボルツマン法を用いた流体シミュレータ

格子ボルツマン法を用いた流体計算処理と可視化処理 (流速の大きさを色として表示する) を GPU で, 抵抗力計算と力覚提示デバイスの制御を CPU でそ

れぞれ行うことによって，物体（円柱）にかかる抵抗力を力覚提示デバイスに提示するとともに，実時間で流体中の物体の位置を変動させることのできる流体シミュレータを実装した [10] .

実装環境は以下に示す通りである .

- CPU:Pentium4 3.2GHz
- GPU:GeForce6800Ultra
- OS:WindowsXP

プログラム言語は C , OpenGL , Cg を用いた .

2.2.1 格子と粒子速度モデル

2次元の格子 BGK モデルとして多く用いられている D2Q9 モデル [16] を用いる . このモデルにおいては，正方形格子で空間を分割し，粒子は静止を含む 9 種類の速度を持つとする . 粒子速度 e_i ($i = 0, 1, \dots, 8$) は，以下のように定義される (図 1) .

$$e_i = \begin{cases} (0, 0) & i = 0 \\ c (\cos [(i - 1)\pi/2], \sin [(i - 1)\pi/2]) & i = 1 \sim 4 \\ \sqrt{2}c (\cos [(2i - 9)\pi/4], \sin [(2i - 9)\pi/4]) & i = 5 \sim 8 \end{cases} \quad (11)$$

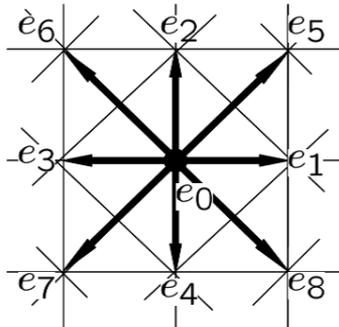


図 1: D2Q9 モデルにおける仮想粒子の速度ベクトル e_i

重み関数 w_i は，以下のように決められる .

$$w_i = \begin{cases} 4/9 & i = 0 \\ 1/9 & i = 1 \sim 4 \\ 1/36 & i = 5 \sim 8 \end{cases} \quad (12)$$

格子ボルツマン法における衝突，並進，巨視化，の一連の過程を 1TimeStep

と呼ぶことにする． K TimeStep 時の時刻は $t = K\Delta t$ となる．

2.2.2 固体壁における境界条件

固体壁における境界条件には，bounce-back 境界条件 [17] というものがある．bounce-back 境界条件では，壁面に向かう粒子は，固体壁に衝突することにより 180° 向きを変えるものとする．全ての速度成分について同様の処理を行うと，壁面上での運動量が 0 になる．すなわち，壁面における粘着条件を表している．本実装においては取り扱う固体壁は円柱であるが，格子に沿った形状に近似した．円柱内部の格子点を固体格子点，円柱外部の格子点を流体格子点とし，壁面は，流体格子点に隣接する固体格子点を結ぶ格子上に存在するものとみなした (図 2) ．

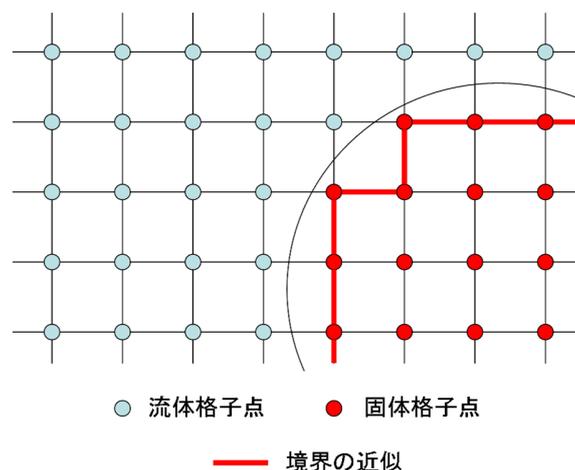


図 2: 固体壁形状の近似

このような固体境界形状の近似であっても，直線状の境界面の移動が境界面に沿った並行移動であれば対応できる [9] ．しかし，固体格子点から流体格子点，あるいは流体格子点から固体格子点への変更が起こる場合には，仮に境界面の移動速度が小さい場合でも，格子点の変更の瞬間，物理変数の急変が起こってしまう．

Bouzidi らは，流体-固体境界面が連続的な任意の位置を取ることを認め，bounce-back 条件を修正することで，非合理的な物理変数の急変を避けつつ境界面の移動を可能にする手法を提案した [11] ．図 3 において，境界面の位置を D とし， A を固体格子点 B ， C を流体格子点としたとき， $q = |BD|/|BA|$ とする．

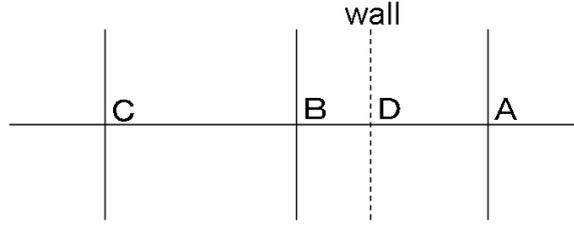


図 3: 格子点と固体境界の位置

i が流体格子点から固体格子点へ向かう向き, i' が固体格子点から流体格子点へ向かう向きを表し, i の向きで, 点 r , 時刻 t での分布関数を $f_i(r, t)$ と表す場合, 未知の分布関数は次式によって求める.

$$f_{i'}(B, t + \Delta t) = 2qf_i(B, t) + (1 - 2q)f_i(C, t) \quad \left(q < \frac{1}{2}\right) \quad (13)$$

$$f_{i'}(B, t + \Delta t) = \frac{1}{2q}f_i(B, t) + \frac{(2q - 1)}{2q}f_{i'}(B, t) \quad \left(q \geq \frac{1}{2}\right) \quad (14)$$

境界が移動する場合, 式 (13)(14) に次の項を付加する.

$$\delta f_{i'}^{(1)} = 2\alpha_i e_i v \quad \left(q < \frac{1}{2}\right) \quad (15)$$

$$\delta f_{i'}^{(1)} = \frac{1}{q}\alpha_i e_i v \quad \left(q \geq \frac{1}{2}\right) \quad (16)$$

ただし, v は境界の移動速度ベクトルであり, D2Q9 モデルにおいては,

$$\alpha_i = \left\{0, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{12}, \frac{1}{12}, \frac{1}{12}, \frac{1}{12}\right\} \quad (17)$$

である.

境界の移動によって固体格子点が流体格子点に変更された場合, 新たな流体格子点では分布関数が定義されていない. 本研究では, v と密度の初期値 ρ_0 によって局所平衡分布関数を求め, これを新たな流体格子点での分布関数とする [12].

固定境界においては, 通常の bounce-back 条件を適用するものとするが, これは式 (14) における $q = 1/2$ の場合に相当する.

2.2.3 GPUによる計算

計算は、GPU上で行う。まず、計算で用いるデータの初期値をCPUで作成し、32bit浮動小数点テクスチャとして、GPUに転送する。テクスチャの各ピクセルと各格子点を対応させる。各格子点が保持する必要があるデータは、9個の分布関数 f_i ($i = 0 \sim 8$)、流速の x 成分 u_x 、 y 成分 u_y 、密度 ρ の計12個である。テクスチャには1ピクセルにつきRGBAの4個のデータを保持することができるため、テクスチャは3枚必要になる。本実装では、固体境界位置情報を保持するためのテクスチャを1枚加え、計4枚のテクスチャを用いた。これらを $\text{tex1} \sim \text{tex4}$ と呼ぶことにする。 f_i の初期値は、 $u = u_0$ 、 $\rho = \rho_0$ によって導出された局所平衡分布関数 f_i^{eq} を用いて、 $f_i = f_i^{eq}$ として与える。 f_i ($i = 1 \sim 4$) は tex1 、 f_i ($i = 5 \sim 8$) は tex2 、 f_0 は tex3 のRGBA 4チャンネルのうちの1つ、たとえばBに格納する。 u_x 、 u_y 、 ρ_0 は tex3 のR、G、Aに格納する。 tex4 ではたとえば流体格子点を0、固体格子点を1とし、RGBAの内の1チャンネルに格納する。 tex4 の残りの3チャンネルは使用しない。

GPUでは、テクスチャマッピング機能を利用し、計算を行う。すなわち、描画の過程でフラグメントプロセッサへの入力としてテクスチャを利用する。フラグメントプロセッサのパイプライン処理をCgによるフラグメントプログラムで制御することで計算を行う。

計算の流れとしては、まず1)円柱面の位置を捉えるための計算を行い、続いて2)衝突3)並進4)巨視化の計算を行い結果を画面に5)表示する。描画回数は、それぞれ1)2回、2)2回、3)2回、4)1回、5)1回であり、計8回の描画は、全て異なるフラグメントプログラムによる。1)から4)については、ピクセルバッファへの描画後、その内容をテクスチャへ上書きすることによってテクスチャを更新する。

2.2.4 触覚デバイス

オペレータによるシミュレーションへのインタラクティブな入出力を実現するために、SensAble Technologies社のPHANTOM Omni [21] (図4)を使用した。オペレータは、このデバイスのスタイラス部を通じてオペレータに力覚を返すことが出来る。PHANTOM Omniの諸元を表1に示す。

表 1: PHANTOM Omni 諸元



図 4: PHANTOM Omni

作業空間	160 × 120 × 70 mm
位置分解能	450 dpi (0.055 mm)
最大摩擦力	0.26 N
最大提示反力	3.3 N
硬直性	X 軸 : 1.26 N/mm Y 軸 : 2.31 N/mm Z 軸 : 1.02 N/mm
慣性	45 g 以下
インターフェース	IEEE-1394 FireWire port

2.3 流体中の物体にかかる抵抗

2.3.1 運動量交換法による力覚提示

格子ボルツマン法において流体中の固体にかかる力を求める手法として Yu らは移動境界問題に対して運動量交換法を使用した [13]。運動量交換法とは、格子ボルツマン法固有の手法であり、実装が容易であるという長所をもつ。そこで我々の初期実装には運動量交換法を用いて力覚の提示を行うことにした。

壁面へ向かう分布関数 $f_R(t)$ で表現される粒子が壁面で跳ね返ることにより $f_L(t + \Delta t)$ で表現されるとすれば、運動量変化 ϕ は、

$$\phi = f_R(t) + f_L(t + \Delta t) \quad (18)$$

となり、これが固体へと渡される。固体にかかる力は、運動量変化全てについて $\phi_i e_i$ の総和をとることで求めることができる。

力の計算は、CPU で行う。円柱表面での粒子の跳ね返りによる運動量変化 ϕ を求めるためには、境界条件適用前と適用後の分布関数が必要となるが、本実装においては跳ね返る前の粒子を 2 個の分布関数で表現している上、どの分布関数が跳ね返る粒子を表現しているかの判断が難しい。ところが、跳ね返りの前後での分布関数の差は式 (15)(16) であるので、跳ね返りの後を示す分布関数と q_i を GPU から読み出せば、 ϕ は計算可能となる。

GPU から CPU へのデータ転送は、数 100MB/s 程度と低速であるため、力を計算する際のデータの読み取りは円柱の存在部分周辺の最小限の領域にとどめられなければならない。

2.3.2 運動量交換法での問題点

運動量交換法を用いて流体のシミュレーションを行った。PHANTOMにより実際に円柱を流体内で動かしてみたところ、確かに流体による物体への抵抗力を感じる事が出来た。しかし、物体の移動を停止した場合に流体からの抵抗力を感じる事ができない等、必ずしもに流体力学的に正しい力がPHANTOMに返されないことが確認された。

そこで、以下のような実験を行った。まず、流速を $U = (U_x, U_y) = (0.01, 0)$ と一定とする。ここでは円柱にかかる力のデータを取るためにPHANTOMからの入力ではなく、プログラムによって円柱の動きを制御する。この一定速度の流体に対し、円柱の移動速度 $V = (V_x, V_y)$ が以下の3つの場合

- 停止： $V = (0, 0)$
(一定流速中に円柱を停止させた場合)
- 順行： $V = (0.01, 0)$
(流体と同じ速度で円柱を流体中を移動させた場合)
- 逆行： $V = (-0.01, 0)$
(流体と逆向きで円柱を流体中を移動させた場合)

を考え、その時に物体にかかる力を計測する(シミュレーション中の流れの様子を図5に示す)。15000TimeStep($V_x = 0.01$ で今回用いたシミュレーション空間を円柱が端から端まで移動するのにかかるTimeStepより短い時間)毎に円柱を以下のように動かした場合のシミュレーションを行った。

- TimeStep=0 ~ 15000:停止
- TimeStep=15001 ~ 30000:順行
- TimeStep=30001 ~ 45000:停止
- TimeStep=45001 ~ 60000:逆行

円柱の直径 $A=16, 24, 32$ とした場合のデータを取った。結果を図6に示す。ここでは、円柱を x 軸方向にのみ動かしているため、円柱にかかる力の x 成分のみを考える。図5を見ればわかるが、流体中に円柱を置くことによって、円柱の上下から交互にカルマン渦と呼ばれるものが発生する。これによって、実際には y 軸方向にも微小な力が働いているはずであり、今回のシミュレーションにおいても実際に y 軸方向の微小な力を観測した。

一般に流体力学では、定常的な流れの流体中にある物体にかかる抵抗力は、流体と物体の相対速度の2乗、物体の基準面積(物体の流れに垂直な平面への投影

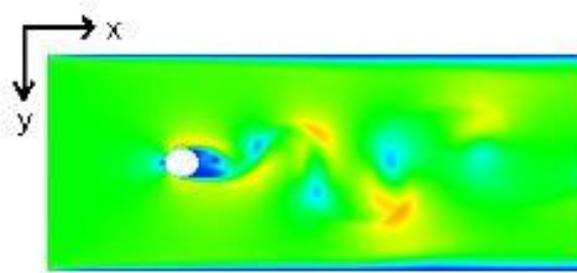


図 5: シミュレーション中の流れ

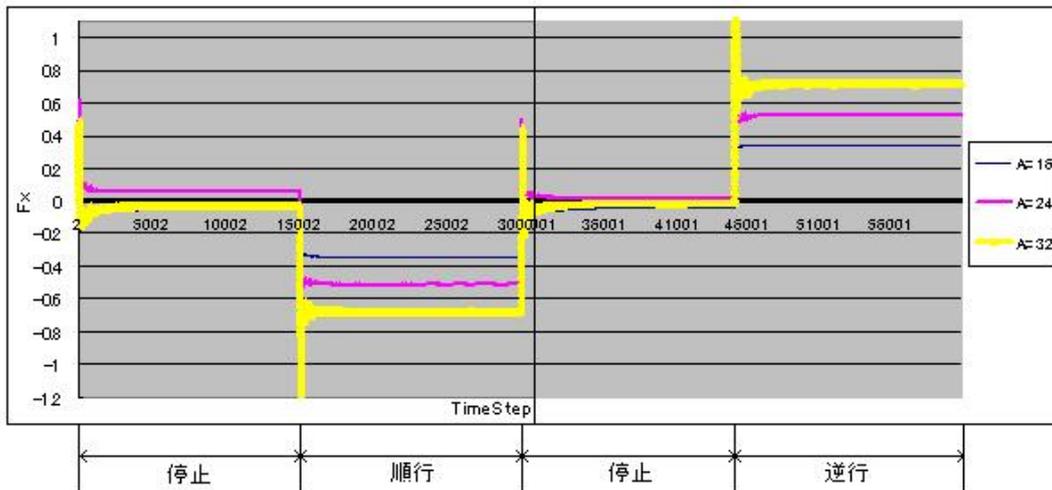


図 6: 円柱にかかる力 (運動量交換法)

面積) に比例するということがわかっている．この関係から，図 6 の結果は，物体が静止している時に物体にほとんど抵抗力が働かず，順行している時は流体と物体の相対速度は 0 であるにも関わらず物体に抵抗力が働いているので，流体力学的に誤りである．

この実験によって，運動量交換法を用いたこの流体シミュレーションにおいて，円柱にかかる力覚をオペレータに正しく提示できていないことがわかった．しかし，ここで図 6 のグラフにおいて，物体が順行状態の時に抵抗力 F_x が 0 となるように上にシフトを行う (図 6)．この図を見ると，相対速度の 2 乗，基準面積に比例してはいないので厳密に流体力学的に正しくはないが，相対速度，基準面積が大きくなれば，抵抗力も大きくなるという傾向は見られた．すなわち，抵抗力の値としては誤りであったが，変位だけを見れば，流体力学的に正しい傾向が示している．PHANTOM により円柱を流体内で動かした時に，流体による物体への抵抗力を感じる事が出来たのは，これが原因であったのではな

いだろうか．

次章では，まず巨視的な流体力学の性質である，相対速度の2乗，基準面積に比例するという性質を正確に表現できる力覚モデルを用いて流体シミュレーションの実装を行った．

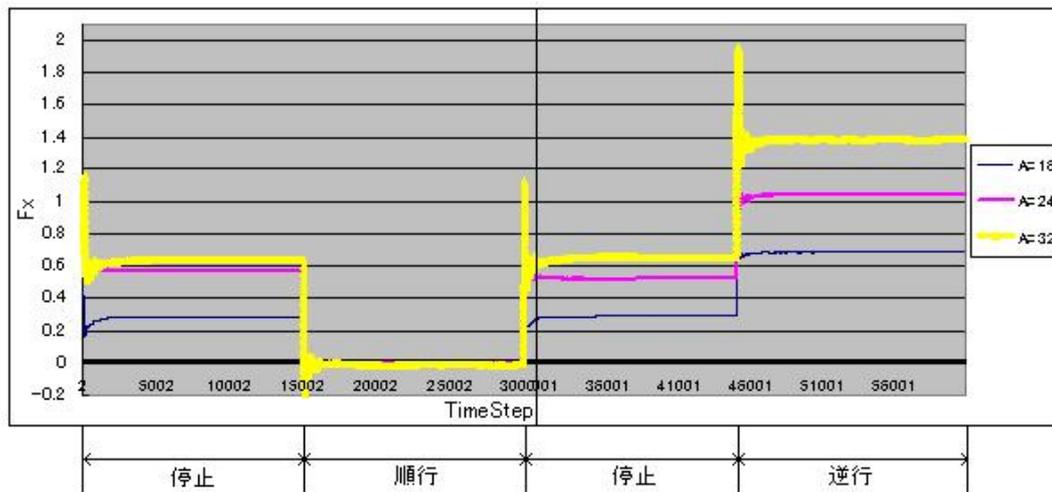


図7: 図6を順行の状態の時に $F_x = 0$ となるように上にシフトしたグラフ

第3章 巨視的流体モデルに基づく力覚提示

一般に，流体力学的抗力はいくつかの要因からなる複雑な現象である．コンピュータシミュレーションにおいて，これらの流体の流れに関する問題を本格的に解析するとなると，実時間でのシミュレーションが困難になってくる．そこで，コンピュータグラフィック系の物理シミュレーションなどで用いられている近似計算モデルを用いて力の計算を行ってみようと考えた [14]．

前章で問題のあった力覚の提示部分に対し，力の計算のモデルを変えてシミュレーションを行い，考察を行った．また，物体の形状を変えた場合についてもシミュレーションを行い，考察を行った．

3.1 相対速度と有効断面積

物体が流体中を運動するときには，その運動を妨げる抵抗力が働く．抵抗力の原因となるものは，速度が小さいときは主として流体の粘性であるが，速度が大きくなると粘性のほかに流体の圧力抵抗が働くようになる．この抵抗力は，

流体の性質によって異なるとともに、運動する物体の形状や速度によっても大きく変化する。

流体による物体への抵抗力を理論的に求めることは非常に困難である。しかし、流体の流れを巨視的に見た場合、流体と物体との相対速度を U 、物体の形状による抵抗係数を C とすると物体にかかる抵抗力は

$$F = C \frac{1}{2} A \rho U^2 = C \frac{1}{2} A \rho (u - V)^2 \quad (19)$$

となることが分かっている。ただし、 u は流速、 V は円柱速度、 A は物体の基準面積である。そこで、力覚の提示を巨視的流体モデルを用いて行ってみることにした。

ここで、問題となるのが平均流速の定義法である。流体が定常的な流れの場合には流速は一定であるので流速を求めるのは難しくない。しかし、本実装では物体をオペレータが動的に動かすことによって流体に流れが生じるために、流れが非定常となり、流速の定義が困難となる。流体による物体への抵抗力は流体の粒子が物体壁面にぶつかる圧力と粒子と壁面との摩擦力によって生じる。そこで、物体に直接影響を及ぼす物体壁面周辺の流速の平均値を流速 u として、物体にかかる抵抗力を計算することにした。

3.1.1 実行結果と考察

円柱にかかる力を式 (19) によって求めた場合のシミュレーションを行った。シミュレーションは 2.3.2 と同様に物体を移動させた場合の物体への抵抗力を測定した。結果を図 8 に示す。

物体が停止している場合は、物体に流体の流れの向きに力がかかり、舞台を順行させた場合物体は、物体には抵抗力はかからず、物体を逆行させた場合は、物体を停止させている場合よりも大きな力がかかる、という一見して正しそうな結果が得られた。これは式 (19) の相対速度の 2 乗に比例するという関係をほぼ満たしていることが確認できた。しかし、基準面積が大きくなると抵抗力も大きくなりはしたが、基準面積に比例するという関係まではこのモデルでは表現できなかった。

以上より、この巨視的流体モデルを流体シミュレータに用いることによって、抵抗力と相対速度との流体力学的関係を満たしている力覚提示が可能となる。

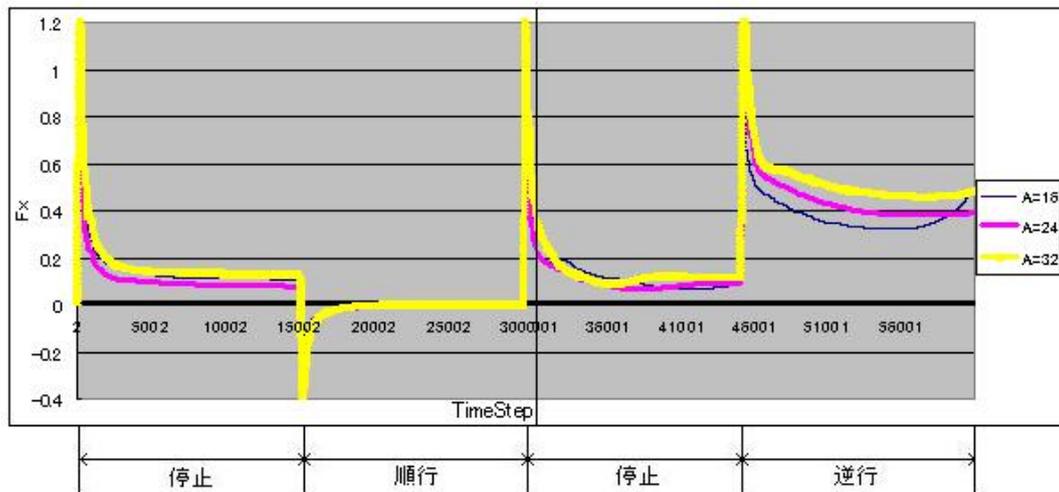


図 8: 円柱にかかる力 (x 成分)

3.2 物体形状の変更

巨視的流体モデルにより，流体中の円柱にかかる抵抗力が相対速度の 2 乗に比例するという関係を満たした力覚提示が可能となった．次に，形状による抵抗係数が既知である図 9 の物体について考える．

3.2.1 形状抵抗

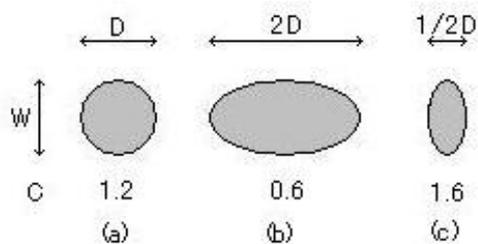


図 9: 物体の形状とその抵抗係数 C

物体形状を図 9 のように変え，先ほどと同様のシミュレーションを行い，物体への抵抗力を測定した．9(b)(c) の結果を図 10 に示す．

さらに，物体の形状が図 11 のような場合を考える．

物体形状を図 11 の 3 つの場合に変え，先ほどと同様のシミュレーションを行い，物体への抵抗力を測定した．結果を図 12 に示す．また，図 13 に一定流速中のシミュレーションを行っている時の結果と，その時の流線を簡単に表した図を示す．

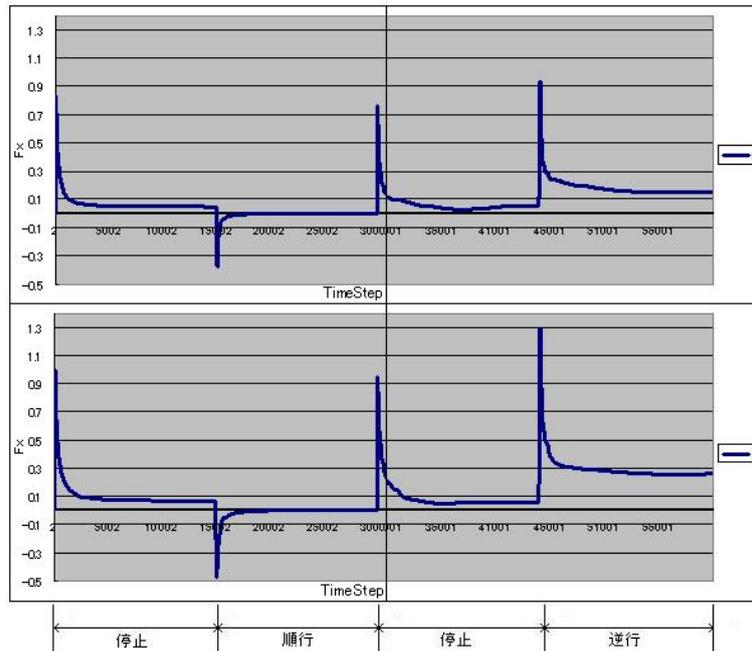


図 10: 図 9 の物体にかかる力 (x 成分) 上 : b , 下 : c

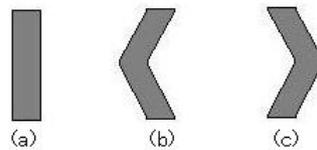


図 11: 物体の形状 1

物体の形状が図 11b (<) の場合は、物体にかかる抵抗力は計測できたが、図 11a (|) , 図 11c (>) の場合では、物体へかかる抵抗力を提示することが出来なかった。この原因として、物体前方の流線は図 13 のようになることから、物体形状によっては物体前方に流体が停滞する領域 (止水域) が出来てしまい、物体周辺の格子点上の流速がほぼ 0 となってしまうため、物体周辺の点の平均流速を流速として力を計算しても正しい結果が得られなかった。

3.3 相対速度

3.2 節より、物体前面に止水域ができるような形状の物体では、物体にかかる力を上手く求められないことがわかった。物体前面の流体は確かにほぼ停止しているが、実際には流体による圧力は止水域の流体を伝わって物体へかかっているはずである。そこで、流速を物体壁面のすぐ近傍の点での流速の平均値を

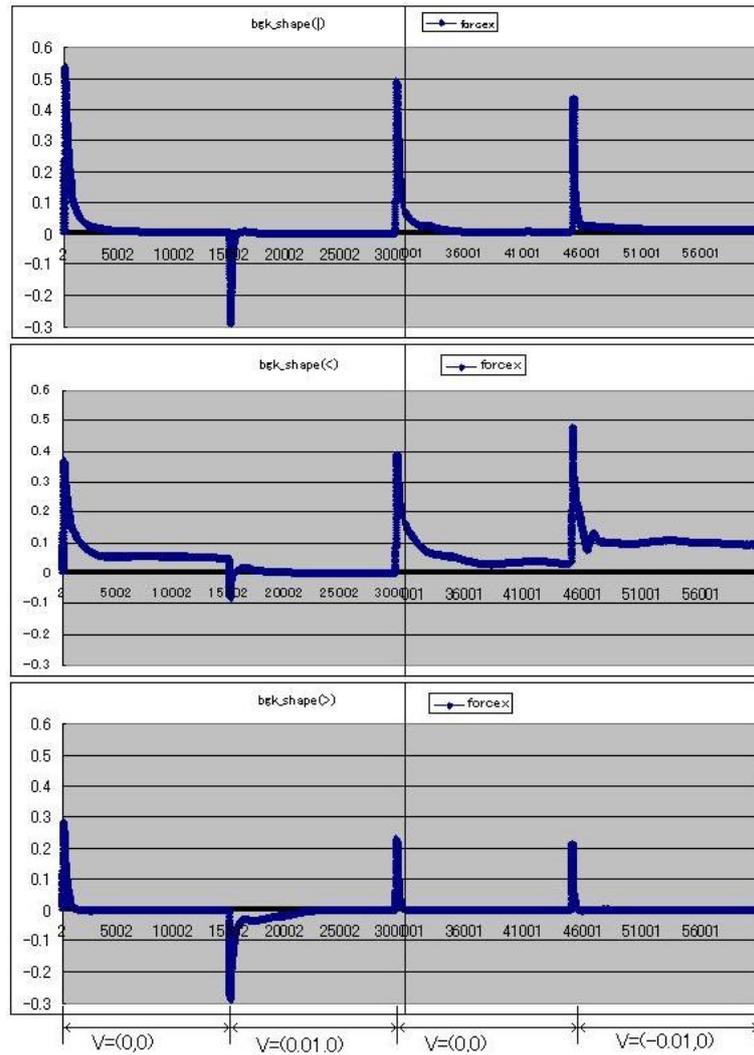


図 12: 図 11 の物体にかかる力 (x 成分) 上 : a、中 : b、下 : c

とるのではなく、止水域の外側の点の流速をとることによって、物体にかかる抵抗力をうまく求められるのではないかと考えた。図 13(a) を見ると、止水域を仮想的な物体の拡張であると考え、物体前面の形状が図 11 のいずれの場合も同じ形になっていることがわかる。そこで、図 11 の形状に関わらず、物体を図 14 のように仮想的に全て同一形状と考え、物体壁面から一定距離 d 離れた点での流速の平均値をとるようにした。図 11(a)(b) において物体前方だけでなく、後方まで図 14 のように流速をとる領域を拡大したのは、流れの向きが変われば、止水域ができる面が変わるからであり、本実装において、定常的な流れではなく非定常な流れに対応するため、このようにした。

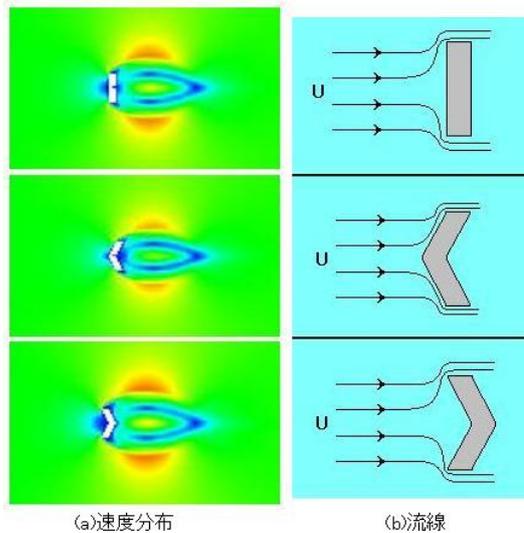


図 13: 一定流速中のシミュレーション

図 14 の d の値を $1, 2, \dots$ と大きくしていき，これまでと同様のシミュレーションを行い，力を計測した．すると， $d = 4$ 以上で物体にかかる力が計測できるようになった． $d = 4$ の時の結果を図 15 に示す．

流速を物体壁面から少し離れた点 (止水域の外側の点) で取ることによって，物体形状に関わらず物体にかかる抵抗力を計測できるようになった．しかし，流れが非定常的な場合にも抵抗力が求められるように，物体後方でも流速をとる領域を物体壁面から離れたために，物体後方にできる流体の渦の影響で，物体形状による抵抗力の違いまでは表現できなかった．物体形状によって適切な抵抗係数を与えてやることによって，この形状による抵抗力の違いは表現できるようにはなるだろう．しかし，円柱のように物体をどの方向から見ても同じ形状の物体と違い，図 11 のような形状の物体は見る方向によって異なる．すなわち，流体の流れの向き，物体を動かす向きによって，力を求める際の抵抗係数，基準面積は変わってしまう．本研究の目的は，インタラクティブなシミュレーションの実現であるため，このままでは正確な力覚の提示を行うことはできない．そこで，計算時間はかかるがより高精度な圧力ベースの計算法の導入を検討する．

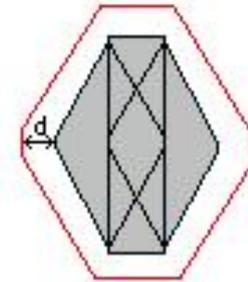


図 14: 流速の取り方

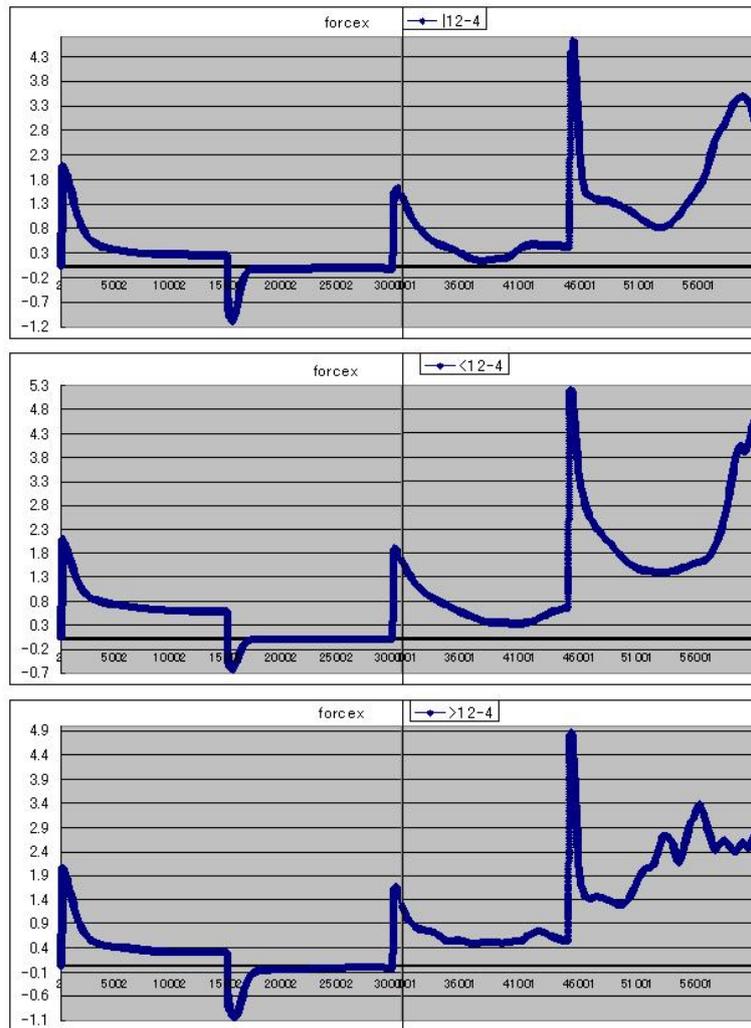


図 15: 流速を図 14 とした時の図 11 の物体にかかる力 (x 成分) 上 : a, 中 : b, 下 : c

第 4 章 物体形状による抗力の違いをを反映した力覚の提示

物体の形状が変われば抵抗力の大きさも当然変わるはずである。物体形状が図 11 のように方向によって基準面積，抵抗係数が変わる場合の，インタラクティブなシミュレーションの実現にはその違いを表現する必要がある。そこで本章では，物体による抵抗力の違いを反映した力覚提示について述べる。

4.1 圧力モデル

流れの中に置かれた物体，あるいは静止した流体中を運動する物体は流体から力を受ける．この力は，物体表面に働く圧力に基づく分布が物体の前後あるいは上下面で非対称なために生じる．物体表面上の任意の微小面積 dA に作用する流体の圧力を p とする．圧力 $p dA$ を物体表面全体にわたって積分することによって，物体にかかる力を求めることができる．

4.1.1 物体周辺の圧力に基づく抗力計算

平衡分布関数が式 (9) で表される時，Chapman-Enskog 展開を用いることによって格子ボルツマン方程式 (4) から

$$\partial_t(\rho u_\alpha) + \partial_\beta(\rho u_\alpha u_\beta) = -\partial_\alpha(c_s^2 \rho) + \partial_\beta(2\nu\rho S_{\alpha\beta}) + \delta\partial_\beta(\tau - \frac{1}{2})\partial_\gamma(\rho u_\alpha u_\beta u_\gamma) + O(\delta^2) \quad (20)$$

の式が導き出せる [18]．これは Navier-Stokes 方程式

$$\partial_t(\rho u_\alpha) + \partial_\beta(\rho u_\alpha u_\beta) = -\partial_\alpha p + \partial_\beta 2\mu(S_{\alpha\beta} - \frac{1}{2}U_{\gamma\gamma}\delta_{\alpha\beta}) \quad (21)$$

と等しい．よって，圧力 p と格子ボルツマン方程式のパラメータの関係は，

$$p = C_s^2 \rho = \frac{1}{3}c^2 \rho \quad (22)$$

となる．格子ボルツマン法において，それぞれの格子点での密度は式 (2) により 1TimeStep 毎に求められているので，それにより物体壁面表面部分の圧力を求めることができる．しかし，物体周りの密度分布を調べてみたところ，初期密度 1 からごくわずかしか変化しておらず，単純に圧力積分を行っただけでは十分な物体への抗力が得られない．そこで Batchelor (1983) らが水の波の表現に用いた状態方程式

$$p = p(\rho) = \beta\left(\left(\frac{\rho}{\rho_0}\right)^\gamma - 1\right) \quad (23)$$

を用いて圧力補正を行う [19]．ここで γ は文献 [20] で野田らが行った実験，ならびに計算速度を考慮して $\gamma = 2$ とする．また ρ_0 は平均密度， β は次元によって調整される定数である．この補正により，密度が平均密度よりも小さい部分の圧力は小さく，平均密度よりも大きい部分の圧力は大きくなる．しかし，この式をそのまま用いて計算を行うと，一つの格子点上の圧力を求めるのに，除算を 1 回，乗算を 1 回，減算を 1 回行わなければならない，計算時間がかかって

しまう．そこで，

$$p = \beta \left(\left(\frac{\rho}{\rho_0} \right)^2 - 1 \right) = \beta' (\rho^2 - \rho_0^2) \quad (24)$$

と ρ_0 の値がそれぞれの格子点で同じ値であることから $\frac{1}{\rho_0^2}$ を外にくくり出すことができる．ここで， ρ_0^2 もあらかじめ求められる値なので，それぞれの格子点において毎回計算する必要はない．よって，圧力を求める際に必要な計算が乗算 1 回に減算 1 回となり，それぞれの格子点上での除算 1 回分の計算量を減らすことができる．これによって，実際の計算時間を図 16 のように巨視的モデルからわずかな計算時間の増加に抑えることができた．

巨視的モデル		57.4423
圧力モデル	高速化前	58.0635
	高速化後	57.6967

図 16: 10000TimeStep の実行時間 (s)

4.1.2 実行結果と考察

圧力による力の計算モデルを用いて，これまでと同様に物体を動かした場合のシミュレーションを行い力を計測した．結果を図 17 に示す．

図 17 より，流体中の物体にかかる抵抗力が相対速度の 2 乗に比例していることが確認できた．また，基準面積にも比例していることが確認できた．

これにより圧力モデルを用いることによって，物体形状に関わらず相対速度の 2 乗，基準面積に比例する関係を満たす抵抗力を求めることができるということが確認できた．

次に物体形状による抵抗力の違いも表現できているか確かめる．図 17 を見ると確かに物体による抵抗力の違いが表現できているように見える．しかし，図 11 の物体の抵抗係数は未知であるため，正確に形状による抵抗力の違いが表現できているかは確かめることができなかった．そこで，抵抗係数が既知の物体 (図 9) でシミュレーションを行い，圧力による力の計算で抵抗係数による抵抗力の違いが表現できているかどうかを調べた．これまでと同様のシミュレーションを行った結果を図 18 に示す．

図 9 より，物体 (b) の抵抗係数は物体 (a) の $\frac{1}{2}$ 倍，物体 (c) の抵抗係数は物体 (a) の約 1.3 倍である．図 18 の結果より，物体 (a) と物体 (b) の関係は抵抗係数

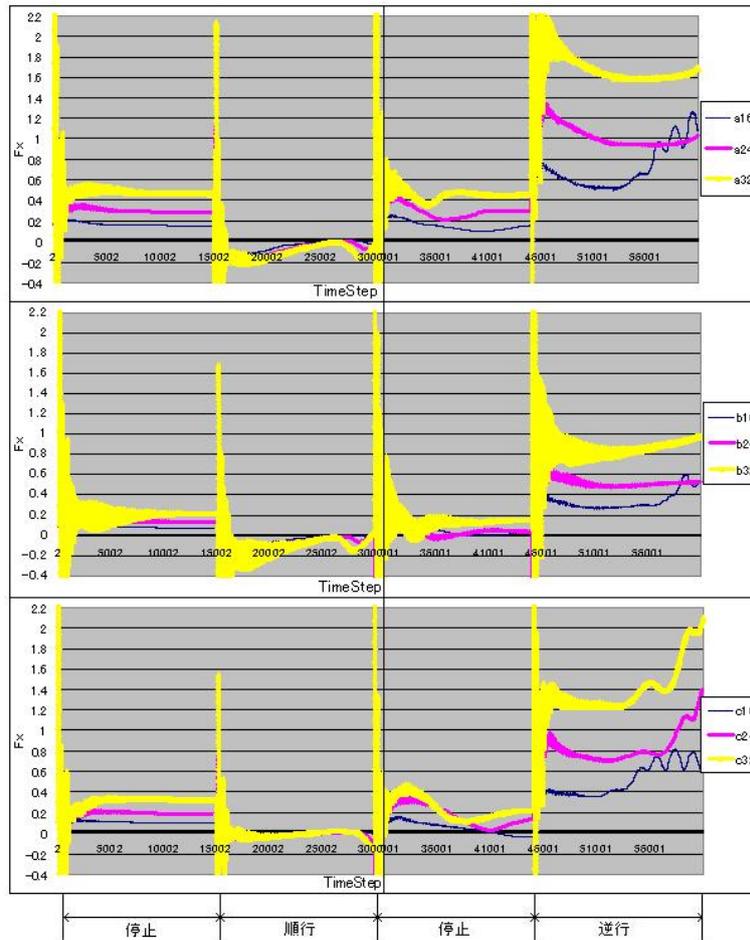


図 17: 図 11 の物体にかかる力 (x 成分) 上 : a, 中 : b, 下 : c

に比例した関係になっていることが確認できた．物体 (c) は物体サイズが小さすぎたためか思うような結果が得られなかった．そこで，物体サイズを大きくしてシミュレーションを行ってみたところ (図 19) 物体 (a) と物体 (c) での抵抗力が抵抗係数に比例した関係になっていることが確認できた．よって，抵抗係数による抵抗力の違いも表現できていることが確認できた．

以上より，圧力による力の計算により，相対速度，基準面積，抵抗係数の関係を満たした流体中の物体にかかる抵抗力を求めることができた．

4.1.3 揚力

物体形状を図 9(b) の下半分を切り取った形状 (図 20) に変えて流速一定 $u = (0.01, 0)$ の流体中に停止させた場合のシミュレーションを行った．シミュレーション中の結果を図 21 に示す．ただし，物体が流体より受ける力を物体中心からのベクトル表示で示す．

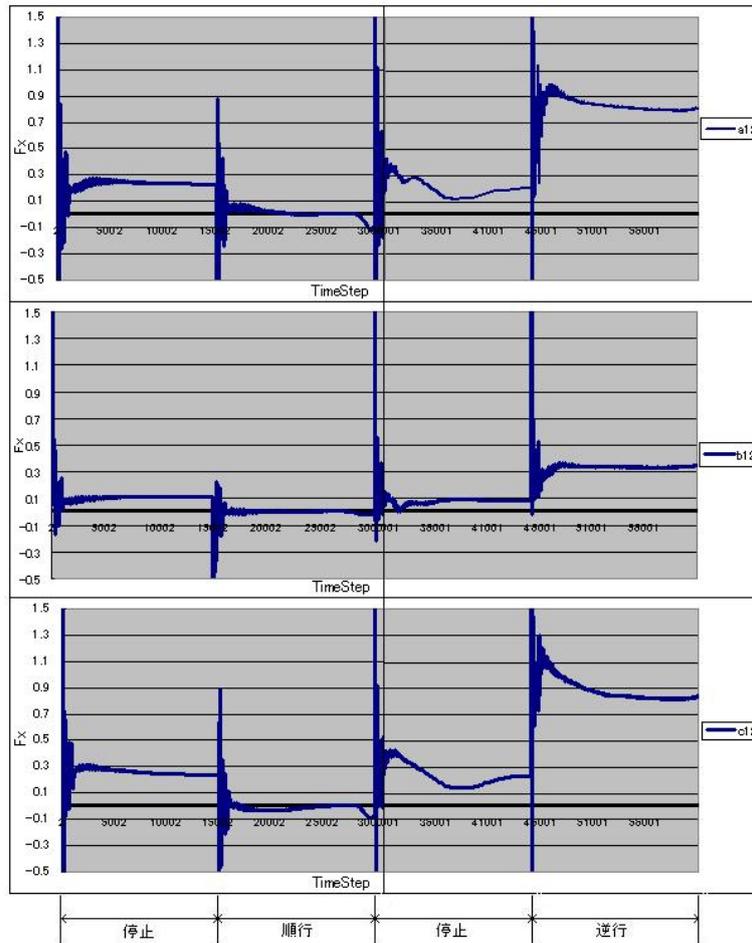


図 18: 図 9 の物体にかかる力 (x 成分) 上 : a, 中 : b, 下 : c

一般に, 20 のような物体では, A 面と B 面では形状が異なるために, A 面と B 面では流速が異なる. これによって, A-B 間に圧力差が生じ, 物体に力が働く. この場合は, B 面より A 面の方が流速が速くなるために, (A 面での圧力) < (B 面での圧力) となり, 物体には B から A の方向に力が働く. このような力を揚力と呼び, 飛行機の翼などが浮力を得るのはこの原理によってである.

図 21 を見ると, 物体には上向きに力が加かかっている, 揚力の提示ができていることが確認できた.

4.2 インタラクティブシミュレーション

流体中の物体にかかる抵抗力を正しく求めることができるようになった. この圧力による力覚の計算モデルによって, 触覚デバイス (Phantom) を用いたインタラクティブ流体シミュレータを改良した. これによって, 運動量交換法を

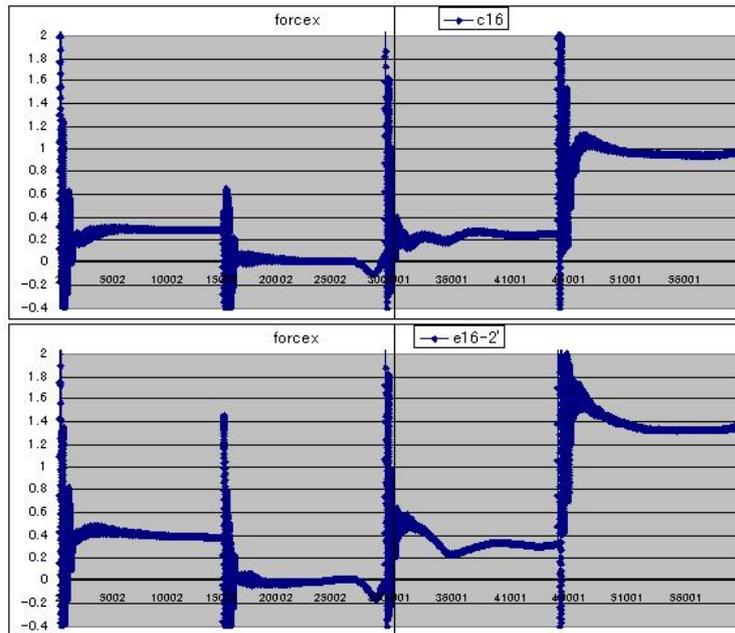


図 19: 図 9 の物体にかかる力 (x 成分) 上 : a , 下 : c

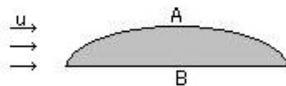


図 20: 物体の形状 2

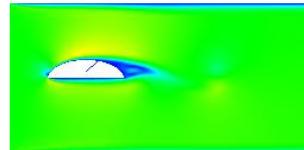


図 21: 図 20 のシミュレーション結果

用いた初期実装では、感じる事ができなかった物体を停止させたときに流体から受ける抵抗力を感じる事ができる、シミュレータを実装することができた。また、物体のサイズ、形状を変えた場合も、流体による物体への抵抗力の流体力学的関係式 (19) を満たした力覚の提示が可能となった。

第 5 章 まとめ

本稿では、まず、格子ボルツマン法を用いた流体計算処理と可視化処理を GPU 上で、抵抗力計算と力覚提示デバイスの制御を CPU 上でそれぞれ行うことによって、物体にかかる抵抗力を力覚提示デバイスに提示するとともに、実時間で流体中の物体の位置を変動させることのできる流体シミュレータを実装した。我々の初期実装で力の計算に運動量交換法を用いた場合には、物体の移動を停止させた場合の力が感じられないなどの力覚の提示における問題点があること

がわかった．そこで，巨視的流体モデルに基づく力覚計算，物体周辺の圧力差による力覚計算を行うことによって，その問題点を改善し，より流体力学的な関係を満たした力の提示が可能となった．この計算モデルを力覚提示が可能なインタラクティブ流体シミュレータに適用することによって，より精度の高い力覚の提示が可能となった．

GPU上で数値計算を行うことで，シミュレーションの高速実行が可能となったが，それでも実時間シミュレーションを行うためにはシミュレーションの空間的制限，流体，物体の速度制限が生じた．GPUの性能向上などにより，さらに高速な計算が可能となり，これらの制限はかなり緩和されるであろう．また，複数のPCで並列計算を行うことにより，より高速な計算が可能となる．

謝辞

本研究の機会を与えて頂いた，本研究室の富田眞治教授に深く感謝の意を表します．また，本研究に関して数々の有用な御指導，御意見を頂いた，嶋田創助手，三輪忍助手、森眞一郎福井大学教授，中島康彦奈良先端科学技術大学院大学教授，に深く感謝致します．さらに，日頃様々な角度から助力して下さいた京都大学大学院情報学研究科通信情報システム専攻富田研究室の諸兄に心より感謝致します．

参考文献

- [1] 富田 眞治, 超高速体感型シミュレーションシステムの研究, 科学研究費補助金, 平成 16 年度基盤研究 (S) 新規採択課題の概要について, 日本学術振興会, p.10 (2004)
- [2] G. McNamara and G. Zanetti, Use of the Boltzmann equation to simulate lattice-gas automata, *Phys. Rev. Lett.* 61, Issue 11, pp.2332-2335 (1988)
- [3] S. Chen and G. D. Doolen, Lattice Boltzmann Method for Fluid Flows, *Annu. Rev. Fluid Mech.* 30, pp.329-364 (1998)
- [4] 蔦原 道久, 高田 尚樹, 片岡 武, 格子気体法・格子ボルツマン法, コロナ社 (1999)
- [5] T. Ertl, D. Weiskopf, M. Kraus, K. Engel, M. Weiler, M. Hopf, S. Rottger and C. Rezk-Salama, Programmable Graphics Hardware for Interactive

- Visualization, Eurographics2002 Tutorial Note(T4) (2002)
- [6] 森 眞一郎, 篠本 雄基, 五島 正裕, 中島 康彦, 富田 眞治, 汎用グラフィクスカード上での簡易シミュレーションと可視化, 電子情報通信学会信学技報, CPSY2004-24, pp.25-30 (2004)
 - [7] General Purpose Computation on Graphics Processor,
<http://www.gpgpu.org/>
 - [8] 小沢 拓, 棚橋 隆彦, 二相系格子ボルツマン法の非構造格子への適用, 計算工学会論文集, No. 20050006 (2005)
 - [9] I. Ginzbourg and D. d’Humières, Local second-order boundary methods for lattice Boltzmann models, *J. Stat. Phys.* 84, pp.927-971 (1996)
 - [10] 小松原 誠, インタラクティブ流体シミュレーション, 京都大学大学院情報学研究科修士論文 (2006)
 - [11] M. Bouzidi, M. Firdaouss and P. Lallemand, Momentum transfer of a Boltzmann-lattice fluid with boundaries, *Phys. Fluids* 13, pp.3452-3459 (2001)
 - [12] P. Lallemand and L. S. Luo, Lattice Boltzmann method for moving boundaries, *J. Comput. Phys.* 184, pp.406-421 (2003)
 - [13] D. Yu, R. Mei, L. S. Luo and W. Shyy, Viscous flow computations with the method of lattice Boltzmann equation, *Progr. Aero. Sci.* 39, pp.329-367 (2003)
 - [14] David M.Bourg, ゲーム開発のための物理シミュレーション入門, オーム社 (2003)
 - [15] 小松原 誠, 森 眞一郎, 中島 康彦, 富田 眞治, 汎用グラフィクスカードを用いた格子ボルツマン法による流体シミュレーション, 情報処理学会研究報告 2005-ARC-163, pp.37-42 (2005)
 - [16] Y. H. Qian, D. d’Humières and P. Lallemand, Lattice BGK models for the Navier-Stokes equation *Europhys. Lett.* 17, pp.479-484 (1992)
 - [17] U. Frisch, D. d’Humières, B. Hasslacher, P. Lallemand, Y. Pomeau and J. P. Rivet, Lattice gas hydrodynamics in two and three dimensions, *Complex Syst.* 1, pp.649-707 (1987)
 - [18] 羽生匡之, 格子ボルツマン法による並列アルゴリズムの開発, 北陸先端科学技術大学院大学情報科学研究科修士論文 (1999)

- [19] H.Gunston and C.H.Batchelor , A comparison of the Priestley-Taylor and Penman methods for estimating reference crop evapotranspiration in tropical countries , *Agric. Water Manage.* 6 , pp.65-77(1983)
- [20] 野田秀希 , SPH 法と分散処理を用いた自由表面を持つ流体の力覚提示 , 福井大学大学院工学研究科機械工学専攻修士論文 (2006)
- [21] SensAble Technologies
<http://www.sensable.com/>