

2003年度 卒業論文

浮遊する剛体の挙動を含めた
水面のリアルタイムシミュレーション

指導教員：渡辺 大地 講師

メディア学部 3DCG アプリケーション構築プロジェクト

学籍番号 00P181

近藤 秀樹

2003年度 卒業論文概要

論文題目

浮遊する剛体の挙動を含めた
水面のリアルタイムシミュレーション

メディア学部
学籍番号：00P181

氏名

近藤 秀樹

指導
教員

渡辺 大地 講師

キーワード

リアルタイムシミュレーション, 水面, 二次元波動方程式,
3DCG, ダンピング係数

コンピュータの処理能力が上がるにつれ、3DCGによる水面のシミュレーションを目にすることが多くなった。水の表現は、環境シミュレーションやゲームシーンに極めて大きな魅力を与える。反射や屈折なども含めて水を表現する方法もいくつか研究されている。しかしながら、浮遊する物体を3次元での傾きも含めてリアルタイムで処理する研究は数が少ない。一般に広く普及しているパーソナルコンピュータ程度の処理速度を想定した、浮遊する物体のリアルタイムシミュレーションは物理的に計算された傾きまでは表現されていないものが多い。そのような擬似的なシミュレーションでは、浮遊する物体を水面の波に関係なく適度に揺らす事で、水面を漂っている様子を表すこともあるが、リアルさに欠けてしまう。

本研究は、一般的なパーソナルコンピュータ程度の処理環境でも、現実性を損なうことなく、浮遊する剛体の挙動を含めたリアルタイムな水面のシミュレーションを行うための研究である。水面の表現に関する計算を単純化し、浮遊する物体の傾きも含めたリアルタイムなシミュレーションを行うことを目的とする。本研究では、水面のみをモデル化し、変形させることで水を表現する手法を採用した。本論文では、水の上に浮遊する物体の挙動の実装手法を提案し、その有用性について考察した。

目次

第1章	はじめに	1
1.1	背景と目的	1
1.2	章構成	2
第2章	水の表現	3
2.1	水をシミュレーションする為の既存手法	3
2.1.1	水面のモデルを用いた手法	3
2.1.2	パーティクルを用いた手法	3
2.2	本研究で用いる手法	4
第3章	浮遊する物体の表現	5
3.1	浮遊する物体の挙動に関する既存手法	5
3.1.1	水の流れを考慮する物体挙動	5
3.1.2	物体の揺れにパターンを持たせる手法	5
3.1.3	物体の傾きを水面の傾きに合わせる手法	6
3.2	物体から水面への影響に関する既存手法	6
3.2.1	テクスチャを用いた手法	6
3.2.2	点を中心に波を発生させる手法	7
3.3	提案手法	7
3.3.1	本研究での前提	7
3.3.2	物体の挙動に関する提案手法	7
3.3.3	物体から水面への影響に関する提案手法	8
第4章	水面の実装	10
4.1	水面のモデルと二次元波動方程式	10
4.2	エッジ部分の操作	13
4.3	波の減衰	14
第5章	浮遊する物体の実装	16
5.1	浮力	16
5.1.1	浮力の大きさ	16

5.1.2	浮力の働く向き	18
5.1.3	浮力の実装	19
5.2	浮遊する物体の回転	20
5.3	浮遊する物体から水面への影響	25
5.3.1	物体の速さによる水面への影響	25
5.3.2	物体からの反射波	26
第6章	評価と考察	29
6.1	評価	29
6.2	考察	30
	謝辞	32
	参考文献	33

第 1 章

はじめに

1.1 背景と目的

近年、コンピュータの処理能力が上がるにつれ、3DCGによる水の表現を目にする機会が多くなってきた。それらは現実での動きを精密に計算するものや、景観としてリアルさを追求したもの等、使用法によってさまざまである。そのような水のシミュレーションの中でも、リアルタイムなシミュレーションを行う場合、物理シミュレーションとしての精度と見た目のリアルさを保つ為、処理速度の向上が求められている。原田ら [3] は、精度を保つには処理にある程度時間をかける必要があるが、処理速度が遅いとリアルさに欠け、精度とリアルさの両立が難しいという大きな問題を抱えていると指摘している。特に水のような流体のシミュレーションにおいて、その精度を高めるためには複雑な演算を行う必要があり、現実での水の動きを忠実に再現するリアルタイムシミュレーションは非常に困難である。そこで、今現在行われている水のリアルタイムシミュレーションでは、水全体の流れを考慮するのではなく、水面だけをモデル化することで計算の単純化を図り、リアルタイム性を保持したシミュレーションを実現するのが主流である。

水面のシミュレーションでは、光の反射や屈折などの水面に波を発生させない現象のシミュレーション [4][6][7] や、物体の落下による波の発生などの水面に動きを与えるシミュレーション [10] も行われている。

水面に動きを与える要素を含んだシミュレーションの一つに浮遊する物体のシミュレーションがある。現在行われている水面のリアルタイムなシミュレーションにおいて、浮遊する物体の挙動はある程度の揺らぎを与えるなどの擬似的な方法[9]を用いる場合が多い。これは、浮遊する物体の挙動を求める為の計算が非常に複雑であり、リアルタイム性を保つことが難しい為である。しかし、現実世界において水面に物体が浮遊する現象は珍しいことではなく、リアルタイムシミュレーションを行う上で、物体の浮遊は重要な現象である。その為、浮遊する物体の挙動を求める計算の単純化と高速化が求められている。

そこで本研究では、水面とそこに浮遊する物体の挙動をリアルタイムシミュレーションによって表現することを目的とし、物体の挙動と物体の挙動による波の発生、物体からの波の反射を表現する為の新たな手法を提案する。提案する手法では、浮遊する物体の挙動は物体の変位に用いる浮力と回転に用いる浮力を別に計算し表現する。別に計算することで、計算を単純化する。物体から水面への影響は、浮遊する物体の移動速度の大きさに比例した波を水面に発生させ、表現する。複雑な計算を行わないことで、計算コストは低くなる。物体からの波の反射に関しては、物体と水面の境目を自由端として波を反射する。

1.2 章構成

本論文では、第二章で水の表現についての既存手法と、本研究で用いる手法を述べる。その後、第三章では浮遊する物体の表現についての既存手法と、本研究での提案手法を述べる。水の表現手法を先に述べるのは、浮遊する物体の挙動に関する提案手法の一部に、水の表現手法の一部を用いた為である。第四章では水面の実装手順を示す。第五章では浮遊する物体の実装手順を示す。第六章では本論文の提案手法を考察する。

第 2 章

水の表現

2.1 水をシミュレーションする為の既存手法

2.1.1 水面のモデルを用いた手法

水のシミュレーションを行う上で、水面を薄い膜のようにモデル化し、モデルを変形させて水を表現する手法 [1][2] がある。モデルの変形には二次元波動方程式によって求めた値を用いる。この手法を用いることにより、水面上でのリアルな波の動きを単純な計算によって行うことが可能である。薄い膜のようにモデル化された水面のモデルはメッシュ状に区切られ、メッシュの各頂点が高さ方向へ移動することによる変形で波を表現する。これら各頂点は、微小領域の高さの変化に合わせて、その近傍は間隔を保とうと、表面張力によって線形なバネの力を及ぼす。水平方向の力は釣り合うので高さ方向にのみ動く。この手法が現在、水のリアルタイムシミュレーションを行うに当たって主流となっている手法である。

2.1.2 パーティクルを用いた手法

水を表現する為の手法にパーティクルを用いた手法 [13] がある。この手法は水全体の流れを計算することが可能である。水の粒子をパーティクルで表現して計算を行う。その為、水面だけでなく水全体のシミュレーションを行うことが可能である。水中のシミュレーションが可能な為、水中を移動する物体や、水流を考

慮した水の表現などを必要とする場合に用いる。ただし、この手法は第 2.1.1 節の手法に比べ、計算コストが高いため、リアルタイムに行うシミュレーションにはあまり適切な手法とは言えない。

2.2 本研究で用いる手法

本研究で用いる手法を選択する上で重要なことは、リアルタイムシミュレーションが可能な処理速度を保つことと、リアルさを失わないことである。

第 2.1.1 節でも述べた通り、リアルタイムなシミュレーションを行うことを目的とした場合、水面を薄い膜のようにモデル化する方法を採用することが主流となっている。これは水面のモデルを用いた手法が処理速度に関して、パーティクルを用いた手法と比較し優れていることが理由の一つである。

二次元波動方程式を利用した水面のシミュレーションには二種類あり、それぞれ陽的な解法、陰的な解法と呼ばれる [2]。水面のリアルタイムシミュレーションで用いられる手法の多くは陽的な解法と呼ばれるもので、陰的な解法よりも処理速度の面で優れている。陽的な解法では、水面を表現するモデルの各頂点と隣接する頂点との距離と比較して、波の伝わる速度を小さく設定する必要がある。条件を満たさなければ水面の高さが指数的に増大してしまう。波の速度を小さく設定した場合、視点を水面に近づけた時のような細かな表現が出来ない。しかし、水のリアルタイムシミュレーションを行う上で、極度に視点を水面に近づける必要のあるシーンは少ない。陽的な解法を用いた場合の波の速度でも、多くのゲームシーンや景観シミュレーションにおける視点と水の距離での水面の表現は十分可能と言える。陰的な解法を用いると波の速度に関する問題は無いが、計算コストが高い為、リアルタイムシミュレーションには向いていない。

以上のことを踏まえた上で、本研究では水流は計算しないものとし、処理速度を重視した結果、水面のモデルを用いた手法における陽的な解法を用いることとする。

第 3 章

浮遊する物体の表現

3.1 浮遊する物体の挙動に関する既存手法

3.1.1 水の流れを考慮する物体挙動

現在、浮遊する物体を表現する手法は、状況によって使い分けられている。物理シミュレーションとして、最も正確といえるのが水の流れを考慮し、水面だけではなく水中の圧力などから物体の挙動を求める手法 [8] である。この手法は、正確な挙動のデータを必要とする海洋科学の分野などで使われる。その為、シミュレーションにおける処理速度はあまり問題ではなく、リアルタイムシミュレーションに向いている手法とはいえない。

3.1.2 物体の揺れにパターンを持たせる手法

浮遊する物体を表現する手法として、物体に対し、水面の挙動に関係なく傾きを与える場合がある [9]。この手法は、ゲームシーンで用いられることがある。物体の傾きは水の動きには影響されず、一定の傾きのパターンを繰り返すだけの表現方法も存在する。この手法の場合、傾きを浮力から求めていないので、浮遊する物体の高さ方向への移動に、浮力の計算を行わない事も可能である。浮力の計算を行わずに、浮遊する物体の高さ方向への変位を実現する場合、物体の高さ方向の位置を水面の高さに合わせる事で実現する手法などがある。

3.1.3 物体の傾きを水面の傾きに合わせる手法

第 3.1.2 節で述べた手法と用いられる状況が似た手法で、浮遊する物体の傾きを水面の傾きに合わせる手法 [9] がある。この手法では、浮遊する物体が存在している点での水面の傾きを利用する為、水のシミュレーションにある程度の正確さが必要となる。

3.2 物体から水面への影響に関する既存手法

3.2.1 テクスチャを用いた手法

浮遊する物体から水面への影響というものはいくつかある。水面の変位に関わる現象や、光の反射による物体の映り込みなどが考えられる。水面の変位に関わる現象とは、水面を伝わる波が浮遊する物体から反射する様子や、浮遊する物体の挙動によって発生する波などである。

これらの表現を行っているシミュレーションの一部に、テクスチャを用いているものがある [10]。テクスチャを用いた手法の場合、波の発生や反射などは水面の変位には影響を与えず、物体の挙動に影響された水面の変位は行われない。例としては、あらかじめ波紋のテクスチャを浮遊する物体の形に合わせて用意し、何らかの条件を満たした場合にのみ浮遊する物体を発生点とした波紋を、テクスチャの拡大を用いて表現する方法がある [10]。また、水面に映りこむ物体の表現もテクスチャを用いる場合がある [9]。しかし、水面に映りこむ物体の虚像を表現する場合、視点の位置によって虚像の見え方も変わる。その為、視点の自由度が高い作品では別の手法で物体の虚像を表現する場合もある。

テクスチャを用いた手法で表現される水面での光の反射は、水面から遠く離れた物体や景色を表現する場合が多い。

3.2.2 点を中心に波を発生させる手法

物体の挙動から発生する波の表現を行う手法には、水のシミュレーションに、直接的な変位を持たせる手法 [2] がある。この場合、浮遊する物体の存在する点を発生源とし、水面に局所的な変位を持たせることで波紋が広がる様子を再現できる。ただし、点を発生源としている為、浮遊する物体が四角い場合でも、広がる波紋は円形となる。

3.3 提案手法

3.3.1 本研究での前提

本研究での水面の表現手法は第 2.2 節で述べた通り、水面のみをモデル化する手法を用いる。水面のみをモデル化するため、水面下の水の変動は考慮されない。さらに水面を構成するモデルの各頂点は、高さ方向への変位のみを行う。つまり、水の横方向への流れを情報として持っていない。以上のことから、水流等を考慮した浮遊する物体の挙動を求めることは困難であると言える。

現実の世界において、浮遊する物体の挙動に影響を与えているのは水から受ける力だけとは限らない。水からの影響以外にも、空気の流れや気圧の変化に影響されている事も考えられる。しかし、リアルタイムでのシミュレーションを行う都合上、計算の複雑化が予想される為、水以外からの影響については本研究では考慮しない。

3.3.2 物体の挙動に関する提案手法

浮遊する物体の挙動に影響を与える力を浮力と重力のみとする。これは、リアルタイムにシミュレーションを行う上で、計算の複雑化を避ける必要があると判断した為である。浮力とは、物体が周囲の流体から受ける圧力の合力である。流体の比重を大きくすることで、流体の圧力も大きくなる。浮力の大きさは、物体が排除した体積の流体の重さに等しい。つまり水に浮遊する物体に働く浮力の大き

きさは、物体が水中に沈んでいる体積分の水の重さに等しくなる。よって、浮遊する物体が沈んでいる分の体積を求めることで、浮力の大きさを導くことが可能となる。

浮力は高さ方向に平行に働くわけではない。浮力のベクトルは水圧の勾配の垂直方向に働く。本研究で用いる水のシミュレーション手法では、水圧を求めることはできない。だが、ほとんどの場面において、水圧の勾配の垂直ベクトルは水面の法線ベクトルとほぼ変わらない [2]。そこで、浮力の働く方向を、物体の存在する水面の法線ベクトルとして計算を行う。

この浮力を用いて、浮遊する物体の挙動をシミュレーションする。本研究で提案する手法では、浮遊する物体を剛体と仮定する。浮遊する物体のモデルの重心を事前に設定する必要がある。これは、ゲームシーンや環境シミュレーション上で浮遊する物体を表現したい場合、その物体の密度が全ての部分において一様とは限らない為である。そして、浮遊する物体の回転は重心を軸に行う。回転は、力のモーメントから回転角速度を求めることで実現する。力のモーメントは、重心から離れた物体上の任意の点における浮力から導く。

3.3.3 物体から水面への影響に関する提案手法

物体から水面への影響に関しては、物体の挙動による水面への影響と、物体から反射する波を実装する。

浮遊する物体の挙動による水面への影響については、浮遊する物体と重なっている水面のモデルの頂点に対し直接、変位を与えることで実現する。物体の重心の速度の絶対値に比例した力を、水面のモデルの各頂点に与える。水面から物体が離れた場合は、影響を考慮しない。物体が水中に沈んだ場合も水面への影響は与えないものとする。

物体から反射する水面の波は、自由端による反射である。波の伝達を波の進行方向と垂直な視点からの断面図で見た場合、自由端による反射は入射波を自由端に対し左右を逆にした波となる。したがって、入射波と反射波は強め合い、自由

端での水面の変位は大きくなる。そこで、物体と水面との境界点を求め、その点に到達した波を自由端として反射すればよい。自由端として反射する手法は第 4.1 節で述べるエッジ部分の操作と同様の手法を用いる。しかし、この手法を用いただけでは水面の変位に合わせて物体の挙動が不安定となる。この不安定性を解決する為に、物体が存在している水面のモデルの各頂点に対し減衰率を高く設定する。減衰率に関しては第 4.3 節で詳しく述べる。

第 4 章

水面の実装

4.1 水面のモデルと二次元波動方程式

水の表現の実装を行うために、水面を薄い膜のようにモデル化する。以下では、これを水面モデルとする。ここで用いる水面モデルは 25×25 に区切られたメッシュ状のモデルを使用する。実際に水面をモデル化した画面を図 4.1 に示す。

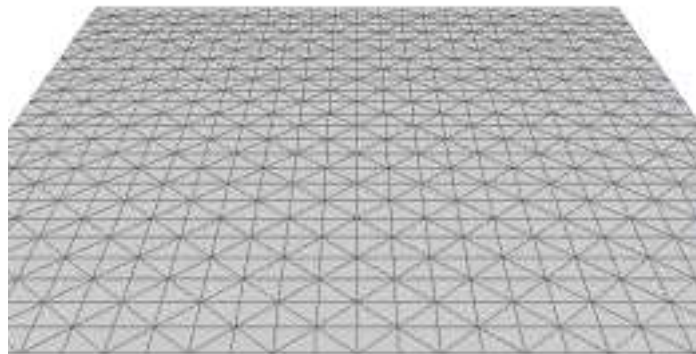


図 4.1: 25×25 に区切られた水面モデル

図 4.1 の水面モデルを構成しているメッシュの隣接する頂点間の距離は 1 とする。これらの頂点を変位させ、水面モデルを変形させることで波を表現する。ここで水面は xy 平面上に広がっており、高さを z 軸方向とする。水面モデルの各頂点の変位計算には二次元の波動方程式を用いる。この二次元の波動方程式は水面を伝

わる波だけでなく、音響の分野でも多く用いられる。実際の二次元波動方程式を(4.1)式に示す。

$$\frac{\partial^2 z}{\partial t^2} = c^2 \left(\frac{\partial^2 z}{\partial x^2} + \frac{\partial^2 z}{\partial y^2} \right) \quad (4.1)$$

ここで c は波が水面上を進む速度である。水面モデルのエッジ部分が上下に動かず、 z 方向への初速度が0の場合、 L を一辺の長さとした $L \times L$ の正方形の領域での一般解を(4.2),(4.3)式 [5] に示す。

$$z(x, y, t) = \frac{2}{L} \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} A_{mn} \sin \frac{m\pi x}{L} \sin \frac{n\pi y}{L} \cos(\omega t) \quad (4.2)$$

$$\omega = \frac{\pi}{L} \sqrt{m^2 + n^2} \quad (4.3)$$

(4.2) 式での係数 A_{mn} を得るための積分計算を(4.4)式に示す。

$$A_{mn} = \frac{2}{L} \int_0^L \int_0^L f(x, y) \sin \frac{m\pi x}{L} \sin \frac{n\pi y}{L} dx dy \quad (4.4)$$

(4.4) 式での $f(x, y)$ は水面の初期状態である [2]。本研究で用いる水面モデルのように、水面が等間隔に並んだ z 値の高さマップとしてモデル化されている場合、(4.4) 式の積分は離散になり、高速フーリエ変換のアルゴリズムを用いて求めることが可能である。 L を一辺の長さとした正方形を領域とした時、実際に水面を近似する為に用いる z 値の高さマップを図 4.2 に示す。

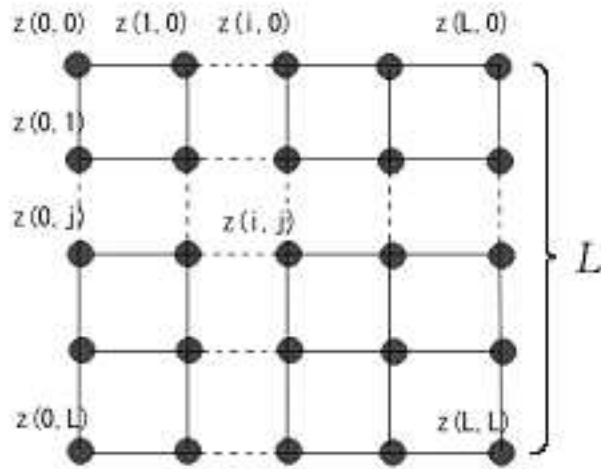


図 4.2: 水面モデルの高さマップ

図 4.2における $z(i, j)$ は、水面モデルの x 軸方向に i 番目、 y 軸方向に j 番目の頂点を表す。さらに h を格子状に区切った隣接する頂点間の距離とし、 N を一辺に並ぶ頂点数とした場合、 $L = h(N - 1)$ で表せる。この級数のうち重要な項だけを計算することで、解の近似は可能である。しかし、この手法は直接的で安定しているが、非常に非効率的である。よって、より効率的な数値解法が必要となる。そこで $z_{i,j}^n$ を x 軸方向に i 番目、 y 軸方向に j 番目の頂点における時刻 t_0 での高さを表すものとして、偏微分を中心差分で近似したものを (4.5) 式に示す。

$$\frac{z_{i,j}^{n+1} - 2z_{i,j}^n + z_{i,j}^{n-1}}{\Delta t^2} = c^2 \frac{z_{i+1,j}^n + z_{i-1,j}^n + z_{i,j+1}^n + z_{i,j-1}^n - 4z_{i,j}^n}{h^2} \quad (4.5)$$

$z_{i,j}^{n-1}$ および $z_{i,j}^{n+1}$ は、それぞれ時刻 $t_{-1} = t_0 - \Delta t$ と $t_{+1} = t_0 + \Delta t$ における高さである。(4.5) 式を $z_{i,j}^{n+1}$ について解いたものを (4.6) 式に示す。

$$z_{i,j}^{n+1} = \frac{c^2 \Delta t^2}{h^2} (z_{i+1,j}^n + z_{i-1,j}^n + z_{i,j+1}^n + z_{i,j-1}^n) + \left(2 - \frac{4c^2 \Delta t^2}{h^2}\right) z_{i,j}^n - z_{i,j}^{n-1} \quad (4.6)$$

(4.6) 式の関係は、単純に $z_{i,j}^n$ の運動が $z_{i,j}^n$ と隣接する点にのみ影響されることを示す。各頂点同士の間隔は一定なので、 h^2 の逆数は事前に計算することが可能である。 h^2 の逆数を事前に計算した場合、乗算と加算、減算が残る。さらに、各頂点での c の値を一定にした場合、全ての係数を事前に計算することが可能である。この方法を用いて、任意の頂点での $t_0 + \Delta t$ における z の値を二回の乗算と五回の加算によって導く。さらに、第二項を省略する為の条件を (4.7) 式に示す。

$$\frac{c^2 \Delta t^2}{h^2} = \frac{1}{2} \quad (4.7)$$

(4.7) 式の条件を満たすような h および c の値を設定することで、計算の簡略化が可能である。

4.2 エッジ部分の操作

ここで言うエッジ部分とは、水面の途切れている部分であり、例えば壁と水面の境界線のことを表す。現実世界での水面のエッジ部分は、通常、高さを固定されていない。(4.6) 式を使った実装方法では、エッジ部分において隣接する頂点がないため不具合が発生する。しかし、エッジ部分に処理を施さなければ、エッジ部分は高さを固定される。その為、エッジ部分にあたる各頂点に適用する式が必要となる。この式を導き出すには (4.6) 式の一部を変更する。右辺第一項の括弧内にある、隣接する頂点の高さ情報のうち、エッジ部分の隣接点として該当しないものを除去する。次に右辺第二項の括弧内にある分数の分子部分にある数値 4 を、演算を行っている点に隣接している点の数に変更する。実際に角にあたる頂点を覗いたエッジ部分で用いる式を (4.8) 式に示す。

$$z_{i,j}^{n+1} = \frac{c^2 \Delta t^2}{h^2} (z_{i+1,j}^n + z_{i,j+1}^n + z_{i,j-1}^n) + (2 - \frac{3c^2 \Delta t^2}{h^2}) z_{i,j}^n - z_{i,j}^{n-1} \quad (4.8)$$

(4.8) 式は図 4.2 に示した高さマップにおける、左端のエッジ部分での式である。その他のエッジ部分に用いる場合には、右辺第一項の括弧内を、それぞれ隣接した各頂点のものに変更する。右辺第二項の数値 4 を、隣接する頂点の数である 3 に変更する。次に高さマップでの角の部分に用いる式を (4.9) 式に示す。

$$z_{i,j}^{n+1} = \frac{c^2 \Delta t^2}{h^2} (z_{i+1,j}^n + z_{i,j+1}^n) + (2 - \frac{2c^2 \Delta t^2}{h^2}) z_{i,j}^n - z_{i,j}^{n-1} \quad (4.9)$$

(4.9) 式に関しても隣接する点毎に右辺第一項、第二項の調整を行う。この場合、(4.7) 式を満たすことによる、(4.6) 式の第二項の省略が行えない為、エッジ部分に関しては右辺第二項の計算が必要となる。

4.3 波の減衰

自然界では、一度発生した波が永遠に水面を伝わるわけではない。波の振幅は時間の経過とともに小さくなってゆく。この現象をシミュレーションで再現する為にダンピング係数 [2] を用いる。ダンピング係数は各頂点の z 値の計算の最後に乗算する。高さマップでの点 $z_{i,j}$ におけるダンピング係数を $d_{i,j}$ であらわした場合の係数の範囲を (4.10) 式に示す。

$$0 \leq d_{i,j} \leq 1 \quad (4.10)$$

ダンピング係数は水面の高さを表す z 値に乘算されるため、ダンピング係数が 0 に近いほど波は小さくなる。よってダンピング係数が 0 に近いほど波の減衰率が高いことになる。ダンピング係数を 1 にすることで、エネルギーの減衰のない自由な動きとなる。

ダンピング係数は海岸線や川などの表現に用いることが可能である。現実世界にある水は、通常、正方形ではない。川や湖、海など傾きが不規則に変化する波打ち際が存在する。堤や岸が水面に対し垂直に近い角度の場合、波は海岸線でほとんどエネルギーを失うことなく反射する。逆に堤や岸の傾きが水面に対し緩やかな場所では、波の反射は非常に弱くなり、肉眼では確認が困難なほど反射が小さくなる場合がある。堤や岸に対し、波が垂直に進まない場合、波は角度を持って反射する。これらの効果は、 z^{n+1} の値を局所的なダンピング係数 $d_{i,j}$ でスケールリングすることによりシミュレートできる。ダンピング係数の値を地形に合わせてスケールリングすることで、波は海岸線等に対してより自然な反応を示す。例えば、堤や岸の傾きが水面モデルに対し垂直に近い場合、ダンピング係数を 1 から 0 に急速に変化させる。このとき水面側が 1 であり、地上側が 0 である。一方、堤や岸の傾きが緩やかな場合はダンピング係数を 1 から 0 に徐々にスケールリングする。特に波の振幅を変化を与える必要のない通常の水面部分では、ダンピング係数を 1 よりもわずかに小さい値に保つ。本研究では、 $d_{i,j} = 0.995$ としてスケールリングする。

第 5 章

浮遊する物体の実装

5.1 浮力

5.1.1 浮力の大きさ

現実の世界では、浮遊する物体は剛体ではない。しかし本手法では計算を簡略化するため、浮遊する物体を剛体と仮定する。このような簡略化は、浮遊する物体が水面波の波長と比較して小さいときには理にかなったものである [8]。まず、浮遊する物体を表すモデルを直方体として実装する。実際に用意したモデルを図 5.1 に示す。

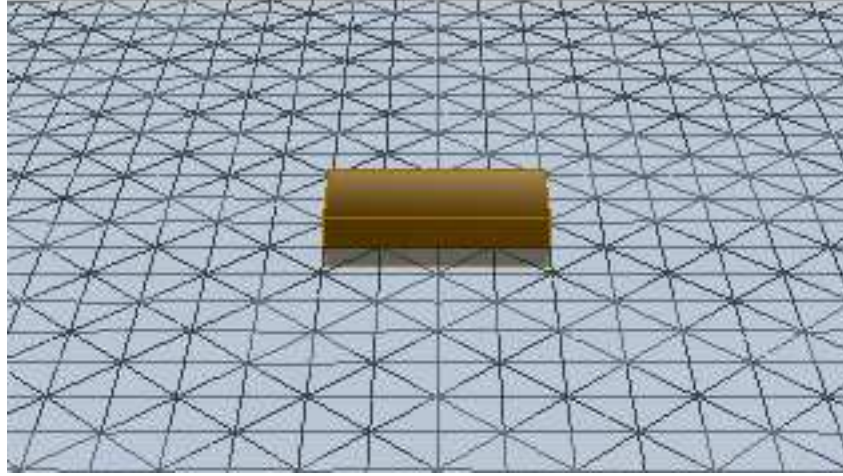


図 5.1: 浮遊する物体を表す直方体モデル

図 5.1の直方体の大きさは、直方体のローカル座標 $(0,0,0)$ を中心とし、 x 方向に 2 と -2、 y 方向に 1 と -1、 z 方向に 0.5 と -0.5 とする。直方体モデルの重心は直方体モデルのローカル座標値 $(0,0,0)$ とする。次に、この直方体モデルを水面上に配置する為に、浮力の計算を行う。浮力は水に沈んでいる体積から求めることができる。直方体の沈んでいる部分の体積は、水面に波が発生している場合、複雑な計算が必要である。複雑な演算を行うことはフレームレートを下げることにつながる為、ここでは計算式を簡略化する。直方体モデルの底面積と物体の重心部分での水面の z 値から浮力の計算を行う。浮力の大きさを F とし、直方体モデルの重心の高さを z_m とし、直方体モデルの重心が存在する xy 座標での水面の高さを $z_{i,j}$ とした場合の実際の計算式を (5.1) 式に示す。

$$F = 8(z_{i,j} - (z_m - 0.5)) \quad (5.1)$$

(5.1) 式の右辺で乗算されている 8 は、図 5.1 の直方体モデルの底面積であり、他のモデルを用いる場合には変更する必要がある。

浮力を求める場合、正確には物体によって排除された体積に、流体の比重を乗算する必要がある。水の比重は一般的に 1 とされているが、実際は水温によって誤差が生じる。水温が 4 °C の場合、水の比重は 1 となる [11]。水温が 0 °C の時の水の比重は約 0.99987 であり、100 °C の時は約 0.95838 となる。その誤差は小さく、自然に存在する水の温度を考慮すると、水の比重は 1 として計算して問題ないと考えられる。

5.1.2 浮力の働く向き

現実の世界において浮力は、水圧の勾配の法線ベクトル方向に働く。水圧の勾配の法線ベクトルは、水面の法線ベクトルとして問題ない程度の誤差である [2]。水面の法線ベクトルは、水面モデルを構成する頂点のうち任意の三点から導くことができる。三点のうち一点から、残りの二点へと伸びるベクトル \vec{A} 、 \vec{B} の外積を求める。外積によって求められたベクトルは、 \vec{A} 、 \vec{B} の両方に対して垂直となる。 \vec{A} 、 \vec{B} に垂直ということは、 \vec{A} 、 \vec{B} によって作られる平面に垂直な為、法線ベクトルとして扱うことが可能である。 \vec{N} を法線ベクトルとした場合の実際のベクトルを図 5.2 に示す。

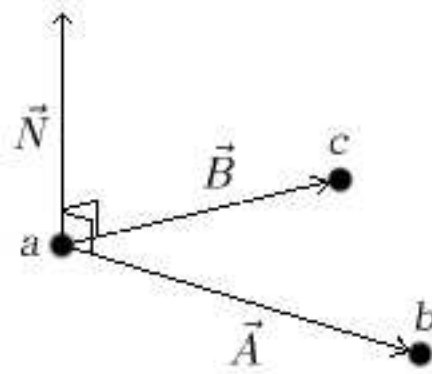


図 5.2: 水面モデル上での法線ベクトル

図 5.2では、水面を構成する任意の 3 点として、 a, b, c として定める。 a から b へのベクトルを \vec{A} とし、 a から c へのベクトルを \vec{B} とする。外積 $\vec{A} \times \vec{B}$ によって導いた \vec{N} は面 abc に垂直となる。 \vec{A} の成分を (A_x, A_y, A_z) とし、 \vec{B} の成分を (B_x, B_y, B_z) とした場合の \vec{N} を (5.2) 式に示す。

$$\vec{N} = (A_y B_z - A_z B_y, A_z B_x - A_x B_z, A_x B_y - A_y B_x) \quad (5.2)$$

外積によって求められる法線ベクトルの向きは、右手座標系においては、 \vec{A} を回転角の小さい方から回転して \vec{B} に重ねる場合の回転方向での右ねじの進む方向となる。

5.1.3 浮力の実装

以上の作業により、浮力の大きさと浮力の向きを算出し浮遊する物体のモデルに与える。この時、浮力は直方体モデルに対して回転の力を加えず、 \vec{N} 方向に平

行移動を行う。浮力のみを実装したシミュレーション画面を図 5.3に示す。

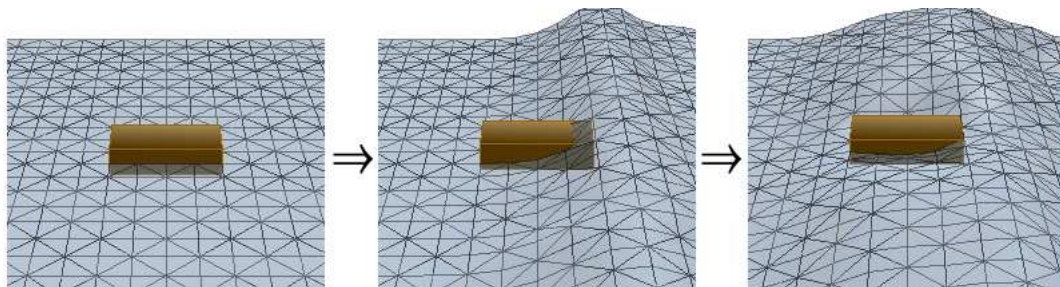


図 5.3: 浮力を重心に加えたシミュレーション画面

図 5.3では回転を行わない為、不自然さを残す。このシミュレーションに直方体の回転を加えなければならない。

5.2 浮遊する物体の回転

浮遊する物体の回転は、現実世界では浮力から求められる。物体が水に触れている全ての境界点における圧力が、浮遊する物体に傾きを与える。本論文での提案手法では、第 5 節で述べた通り、浮遊する物体を表す直方体モデルの重心に浮力を与え、物体の水面浮遊を実装している。ここでさらに、直方体の各質点における浮力を求め、直方体に力として加えることは不自然であり、計算を難解にする。そこで、本研究の提案手法では、直方体の任意の 4 箇所に、物体の傾きを求めるための頂点を設定する。以後、この頂点を力のモーメントの作用点とする。実際の作用点を図 5.4に示す。

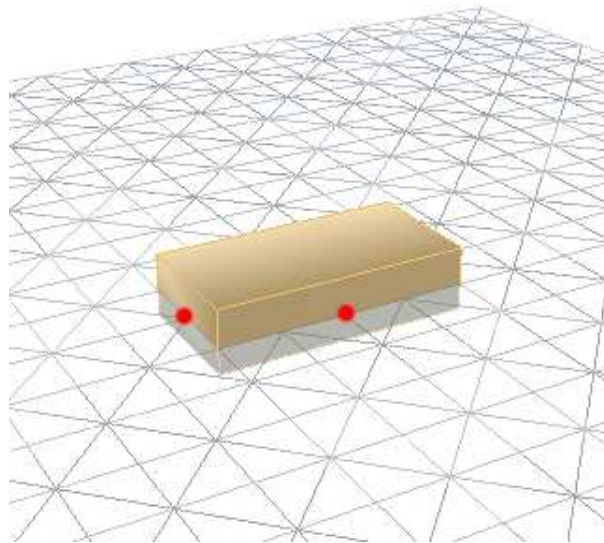


図 5.4: 物体の傾きを求めるための点

図 5.4では、直方体の裏側に配置された作用点は隠れている。この、各作用点の xy 座標における水面の高さと、各質点との差から、水と物体の端との接触の有無を判定する。水面に触れている、もしくは作用点の z 値が水面の高さより低い場合のみ、浮力の大きさを計算する。しかし、浮力の大きさを (5.1) 式と同様に求めると、物体の挙動が不自然となる。原因は参照点における水面との接触の有無の判定が行われた時点での浮力の大きさが、突然大きくなることである。直方体モデルと水面の断面図における浮力の働く面積を図 5.5に示す。

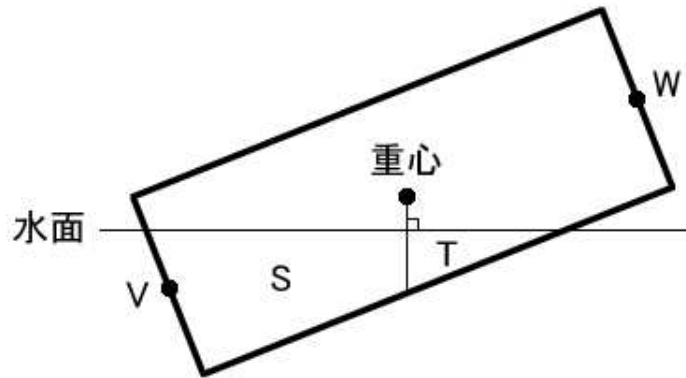


図 5.5: 浮力計算の断面図

図 5.5における S および T は、浮遊する物体によって排除された水の領域である。 V および W は作用点を表す。 S の領域の体積から受ける浮力は直方体を時計回りに回転させようとする力を発生する。同様に T の領域の体積は逆時計回りの力を発生する。ところが、作用点 W は水面から離れている。つまり、本来は逆時計回りの力のモーメントを受けているはずの直方体に対し、作用点 W では力のモーメントを加えていない。これが、浮遊する物体を表す直方体モデルの挙動が不自然になった理由である。

S および T の領域の体積を求めることで浮力の大きさは求めることができるが、浮力を力のモーメントとして受ける点は作用点ではない。物体の傾きによって、力のモーメントが加わる点は移動する。この点をシミュレーションの各フレーム毎に求めていた場合、計算にかかるコストが大きくなってしまい、フレームレートの低下につながる。そこで、安定した動きを実現する為に、各作用点を中心とした球体を配置し、球体にかかる浮力から直方体の傾きを制御するものとした。実際に球体をモデルとして表示した画面を図 5.6 に示す。

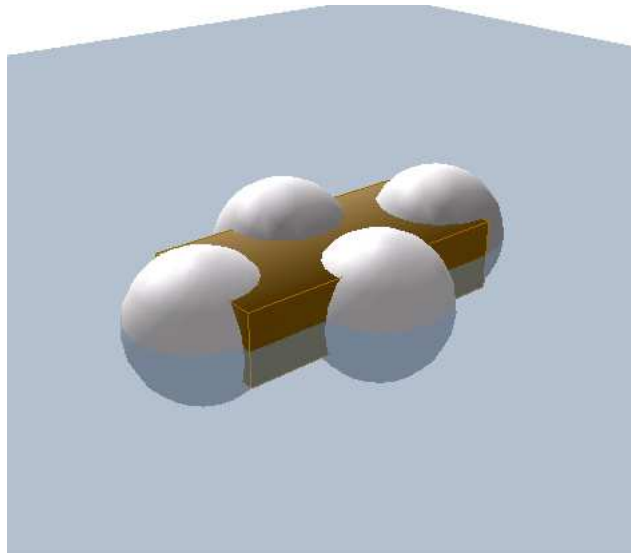


図 5.6: 回転の浮力を求める為の球体

球体ならば直方体の傾きによって、浮力の大きさ、力のモーメントがかかる点
が変動することは無く、作用点と水面との距離を求めるだけで浮力の大きさを求
めることが可能となる。浮力の大きさについても、直方体の傾きによって急激に
変わることが無く安定した挙動を期待できる。傾いた物体と、浮力を求める為の
球体の断面図を図 5.7に示す。

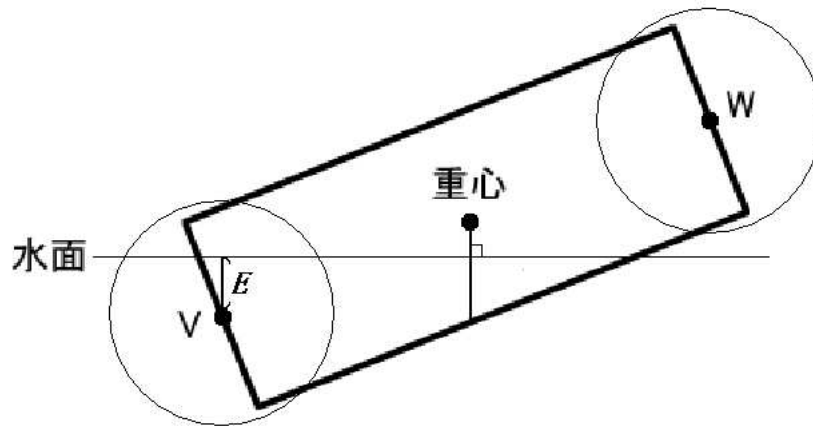


図 5.7: 回転の浮力を求める為の球体の断面

図 5.7における E は、作用点 V の z 値と、作用点 V の存在する xy 座標での水面の z 値との差である。作用点 V を中心とした球体にかかる浮力は、 E を求めることで算出する。

力のモーメントによって直方体を傾ける場合、二つの回転角速度を求め、それぞれを干渉させずに行う。四箇所の作用点を V, W, V', W' として、実際に用いたモデルを真上から見た図を図 5.8 に示す。

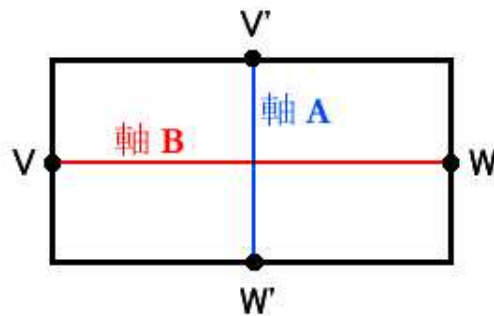


図 5.8: 浮遊する物体の回転軸

図 5.8 中の軸 A 、軸 B はそれぞれ作用点を結んでいる直線ではない。各作用点

の xyz 座標値を $\mathbf{V} = (V_x, V_y, V_z)$ 、 $\mathbf{W} = (W_x, W_y, W_z)$ 、 $\mathbf{V}' = (V'_x, V'_y, V'_z)$ 、 $\mathbf{W}' = (W'_x, W'_y, W'_z)$ とした場合、軸 A は、高さが直方体の重心の z 値に等しく、二点 $(V'_x, V'_y)(W'_x, W'_y)$ を通る直線である。同様に軸 B は、高さが直方体の重心の z 値に等しく、二点 $(V_x, V_y)(W_x, W_y)$ を通る直線である。作用点 V および W で求めた力のモーメントによる回転は軸 A を回転の軸として行い、作用点 V' および W' で求めた力のモーメントによる回転は軸 B を回転の軸として行う。実際にシミュレーションを行った画面を図 5.9 に示す。

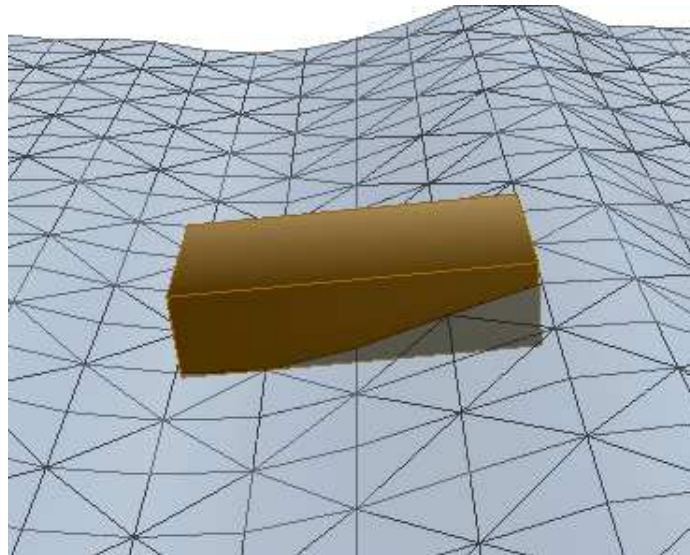


図 5.9: 傾きのシミュレーション画面

5.3 浮遊する物体から水面への影響

5.3.1 物体の速さによる水面への影響

本手法では、浮遊する物体から水面に与える影響の一つとして、物体の速度に合わせて水面に波を発生させる。本来、物体から水面、または水に与える影響は複雑である [12]。物体が動いた場合、水は力を受け、流れを発生する。しかし、水面のみをモデル化している本研究の水の実装手法では、水流を考慮することが難しい。そこで、本論文の提案手法では、水面モデルの各頂点に対し、直接変位を

与える。

与える変位の大きさは、水面に触れている物体の速度の絶対値に比例させる。物体の速度を \vec{V} とした場合の変位の大きさを求める式を (5.3) 式に示す。

$$z_{i,j}^{n+1} = z_{i,j}^n - \alpha |\vec{V}| \quad (5.3)$$

(5.3) 式における α は適当な定数を用いる。水面に与える変位の方向は $-z$ 方向である。物体が水面を浮遊している場合、浮遊する物体は水に対して、圧力とは逆方向に力を加えている。現実世界において浮遊する物体が水に与える力の合力は、浮力の働く方向の逆ベクトルである場合がほとんどである。本来は、直方体の側面が水に与える力は側面の法線ベクトル方向であるが、側面の法線ベクトルは、シミュレーションの大部分のシーンで xy 平面に対し平行に近い。水面モデルの各頂点に対し、 z 軸方向以外の変位を実装するには、水の流れを考慮した手法でなければならないので、今回は与える変位を $-z$ 方向のみとした。

変位を与える頂点は、直方体モデルと重なっている頂点に限る。重なる頂点の算出は、各頂点の位置情報と物体の位置情報を比較すればよい。

5.3.2 物体からの反射波

水面を伝わる波が垂直な壁に反射する場合、壁は上下振動に対して自由端である。浮遊する物体についても、物体の傾きによって波がエネルギーを失うことはあるが、固定端ではない。そこで、浮遊する物体を自由端として波の反射を実装する。自由端における反射については、第 4.2 節で述べたエッジ部分の操作と同様の実装手法を用いる。自由端として扱う水面モデルの頂点を図 5.10 に示す。

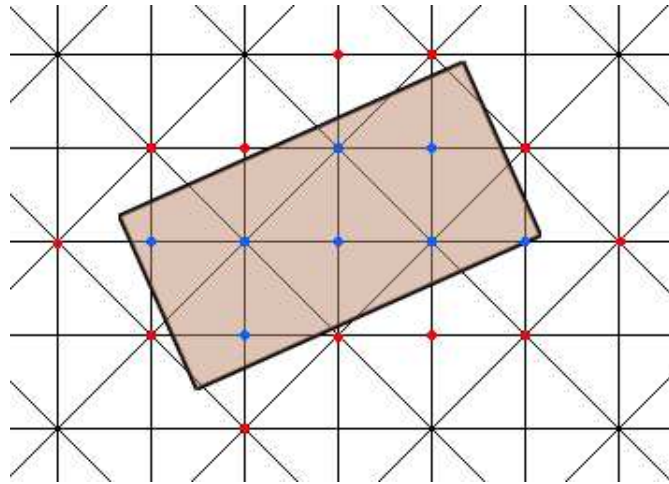


図 5.10: 自由端として扱う水面モデルの頂点

図 5.10で青い点は、直方体モデルと重なっている頂点である。赤い点が、エッジ部分と同様の式を用いる頂点である。エッジとして扱う各頂点では、隣接する頂点のうち直方体モデルと重なっている頂点の情報を考慮しないように調整を行う。ただし、エッジとして扱う各頂点のダンピング係数を 0.995 より小さく設定する必要がある。浮遊する物体から反射する波は全てのエネルギーを反射しているわけではない [12]。実際には、浮遊する物体は水底に固定されておらず、浮遊する物体の下方には水が存在する。当然そこには水の粒子の流れが存在し、波を物体の反対側に伝える。

そこで本手法では、物体からの反射波をダンピング係数によって調整する。さらに、浮遊する物体と重なる各頂点のダンピング係数も 0.995 より低くスケールリングする。これらのスケールリングの値は、想定する物体の質量によって調整する。質量が大きいほど、物体と重なっている頂点での $d_{i,j}$ を小さくする。本研究では、物体と重なっている頂点に対して $d_{i,j} = 0.85$ としてスケールリングする。こうすることで、波のエネルギーの一部を浮遊する物体の反対側に伝えることができ、リアルな表現を可能とする。

実際に提案手法によって、浮遊する物体から水面への影響を実装した場合のシ

シミュレーション画面を図 5.11に示す。

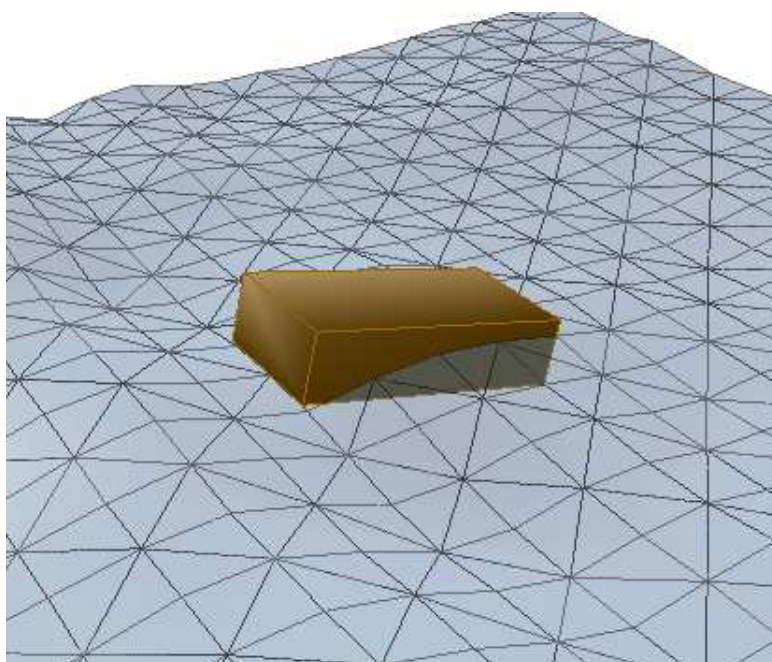


図 5.11: 水面への影響を考慮したシミュレーション画面

第 6 章

評価と考察

6.1 評価

本研究でのシミュレーションは nVIDIA GeForce4 チップセット環境で行った。CPU は Celeron1.70GHz、メモリは 512MB を使用した。

最終的に水面モデルの頂点の数を 100×100 に増やし本手法でのシミュレーションを行ったところ、平均フレームレートを約 30fps 前後に保つことができた。実際に 100×100 の水面モデルを使ったシミュレーション画面を図 6.1 に示す。

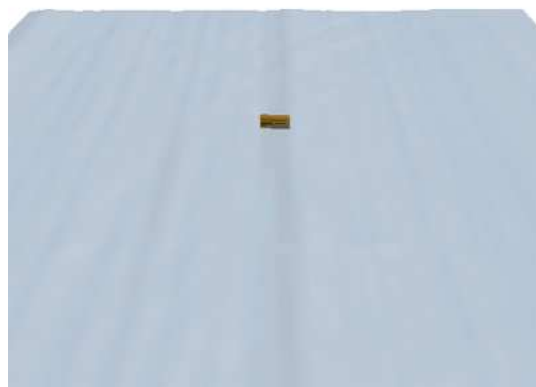


図 6.1: 100×100 の水面モデルを用いたシミュレーション画面

物体から水面への影響の表現も実装できた。物体が移動したときの波の発生

表現も、物体の移動速度から実装できた。浮遊する物体が移動した場合の、波の発生を表したシミュレーション画面を図 6.2 に示す。

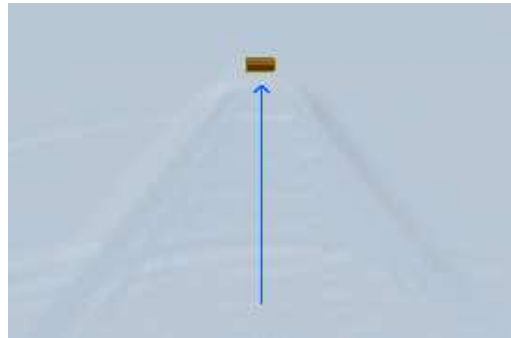


図 6.2: 浮遊する物体の移動による波の発生

本研究での提案手法による物体の挙動のシミュレーションは、処理速度とリアルさに関して問題ない程度の表現を実装できた。

6.2 考察

物体からの水面波の反射は、波の波長に対する物体の大きさによって無視できる程度の大きさになることがある。現実世界で木の板を浮かべた水面に波を起こしたところ、それほど大きな反射波は確認できなかったが、海洋建造物などの大きな物体を表現する場合には、反射波を表現しないことは不自然だと考える。反射波の実装手法に関しては今後も検討する必要がある、場合によっては、表現したいシーン毎に、実装手法を変更する必要がある。

物体の挙動による水面波の発生は、あまり大きな動きを持たせなければ、自然といえる挙動を実現できた。しかし、二次元波動方程式を用いた水面による制限を受けた為、物体が水を掻き分けて進む様子などに不自然さを残した。さらにリアルさを追求するためには、波を掻き分けて進む場合の水面モデルの変形に関する、新たな手法の提案が必要となる。

本研究で提案した手法は、正確なシミュレーションとは言えないがリアルタイ

ムな処理を十分可能としている。今後、コンピュータの処理能力の向上につれ、本研究で用いた既存手法である水面のみをモデル化した手法ではなく、水全体の流れも考慮したリアルタイムシミュレーションも可能となることを期待する。

謝辞

本論文を締めくくるにあたり、ご指導頂き適切な助言を頂きました、本校メディア学部の渡辺大地講師および和田篤氏（電気通信大学）をはじめ、助言をくださったすべての方々に心より感謝申し上げます。

参考文献

- [1] Jens Schneider and Rudiger Westermann, "Towards Real-Time Visual Simulation of Water Surfaces", Proceedings of the Vision Modelling and Visualization Conference 2001, pp.211-218, 2001.
- [2] Mark DeLoura, "GAME PROGRAMMING GEMS", ボーンデジタル, pp180-187, 2001.
- [3] 原田均, "CML を用いた 3 次元ボリュームレンダリングによる水のシミュレーション", 2002.
- [4] nVidia Developer relations,
http://developer.nvidia.com/page/cg_main.html .
- [5] 大槻義彦, "div,grad,rot,..." , 共立出版, 1993.
- [6] OPEN-PROGRAMMING,
<http://homepage1.nifty.com/open-prog/indexold.htm> .
- [7] Kamiyan ' s Homepage,
<http://www.asahi-net.or.jp/~qs7e-kmy/study2/index.html> .
- [8] 谷澤克治, "自由表面を含む流力弾性問題の非線形シミュレーション法", 九大応力研共同研究集会梗概集, Vol.2, 1999.

- [9] 流体力学研究所,
<http://homepage2.nifty.com/fluidmechanics/directx.htm> .
- [10] Kano's webpage,
<http://cgi3.tky.3web.ne.jp/~tkano/tlwater.shtml> .
- [11] JR2BVB,
<http://www4.ocn.ne.jp/~katonet/kagaku/water.htm> .
- [12] 前田久明, "浮遊海洋構造物と流体との相互干渉について", 日本流体力学会和文機関誌「ながれ」第20巻, pp285, 2001.
- [13] 越塚誠一, "数値流体力学", 培風館, pp151-194, 2001.