適応的データ構造を用いた気泡のシミュレーション

上　田　卓　也[†]　　藤　澤　　　誠[†]　　三浦　憲二郎[††]

映画等で用いられているリアリティを追求した CG アニメーションでは，炎や水などの気体や流体の運動を表現するためにコンピュータによる物理シミュレーションが主に用いられている．しかしながら，水のように粘性の低い流体に対しても気泡が扱われておらず，現実の水と比べ粘性の高い液体に見えてしまう．そこで，流体シミュレーションのリアリティをより高めるために，流体中での気泡の動きや変形を扱うことのできる，気泡を考慮した流体シミュレーションを行うことを本研究の目的とする．

そこで，本研究では一様に同じ大きさの計算格子を用いるよりも，データ量や計算時間で有利と考えられる適応的データ構造を用いた計算格子を用い，気泡の周りのみ計算格子を細かくする．このように作成した計算格子を用いて圧力場と速度場を数値計算により求め，それらを用いて気泡の速度を更新していくアルゴリズムを開発した．一様に細分割した計算格子を用いたシミュレーションと比較して，気泡の周りのみ計算格子を小さくした場合には，必要とするメモリが約 10 分の 1，計算時間が 16 分の 1 となった．

Bable Simulation with An Adaptive Data Structure

Takuya Ueda,[†] Makoto Fujisawa[†] and Kenjiro T. Miura[††]

In order to create realistic animations of computer graphics for movies and other commercial films, the physically-based simulation is mainly adopted to realize gas or liquid flows for rendering flames and water movements. However no babble is dealt even for simulation of a low viscosity liquid like water and it looks like a high viscosity liquid rather than the real water. Hence the purpose of this research is to realize a simulation of fluid with babbles that can treat their deformations as well as movements.

In this research, we use non-uniformly subdivided grids with an adaptive data structure instead of uniformly subdivided grids to decrease the amount of data and the computation time required to guarantee the same accuracy, and we make the size of the grids around babbles smaller than other parts. We compute pressure and velocity field with using the adaptive grids generated in such a way and develop an algorithm to update babbles' velocities by use of them. Compared with the simulations with the uniformly subdivided grids, those with the adaptively subdivided grids require about one tenth memory size and one sixteenth computation time.

1. は じ め に

映画等で用いられているリアリティを追求した CG アニメーションでは，炎や水などの気体や流体の運動を表現するためにコンピュータによる物理シミュレーションが主に用いられている．しかしながら，水のように粘性の低い流体に対しても気泡が扱われておらず，現実の水と比べ粘性の高い液体に見えてしまう (図 1 参照).

そこで，流体シミュレーションのリアリティをより高めるために，流体中での気泡の動きや変形を扱うことのできる，気泡を考慮した流体シミュレーションを行うことを本研究の目的とする．

本論文では，まず第 2 章で予備研究として行った速度場によって単純に流される気泡のシミュレーションについて述べる．次に，第 3 章で適応的データ構造による計算格子とその生成について述べ，第 4 章で圧力場・速度場の計算方法について，第 5 章で結果を述べる．最後に第 6 章で研究をまとめる．

2. 予 備 研 究

予備研究として，気泡の変形を考慮せずに単純に速

† 静岡大学大学院理工学研究科
Graduate School of Science and Technology, Shizuoka University
†† 静岡大学工学部機械工学科
Department of Mechanical Engineering, Shizuoka University

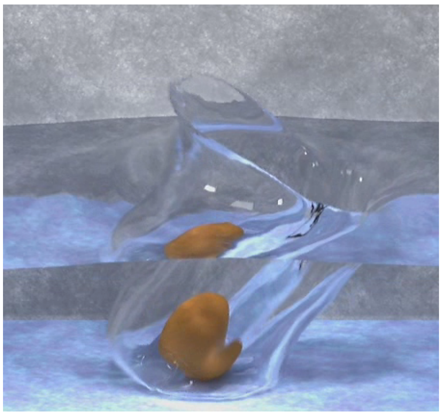


図 1　気泡の無いシミュレーション

度場によって流される気泡のシミュレーションを行った．このシミュレーションで用いた計算格子は 64×64 の直交の構造格子で，圧力と速度がセル中心で定義される．

$$\nabla \cdot \mathbf{u} = 0 \tag{1}$$

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

シミュレーションの処理時間を削減するために，式 (2) のナビエ・ストークス方程式の解法として，タイムステップを大きくしても安定で計算値が発散しないセミラグランジュ“ 安定流体 ”法 [1], [2] を用いた．これは移流項の計算において用いられる手法で，点 x，時間 t での速度場を元に点 x' までバックトレースし，周りの点から線形補間などによって求めた点 x' での速度を点 x，時間 $t + \Delta t$ での速度をとする方法である．

非圧縮性流体を考える場合，式 (2) が非線形項を含み，時間微分と空間微分が混在することから，圧力項において式 (1) の連続の式を満足させ体積流量を一定に保つために，式 (3) で定義されるポアソン方程式を用いて圧力 p を求めるプロジェクション法を用いた．

$$\nabla^2 p = \frac{\nabla \cdot \mathbf{u}^*}{\Delta t} \tag{3}$$

拡散項においては 2 次の中心差分を用い，陰解法で解いた．

結果は，気泡による気泡周りの流体に乱れが発生せず，実際の流体中の気泡の動きに比べて気泡の動きが単純であった．このシミュレーションにおいて，気泡は体積を持たず，仮の (速度場の計算においては考慮されていない) 体積による浮力と気泡の中心位置での速度場によって流されると仮定しており，速度場から気泡への影響はあるが，気泡から速度場への影響はない．したがって，気泡の動きが単純であった理由は，気泡による気泡の周りの流体への影響を考慮していないことによると考えられる．気泡による気泡の周りの流体への影響を考慮すると，図 3 における矢印のような気泡の周りを巻き込む流れの発生が期待でき，またその流れによって気泡の動きも影響を受けると考えられる．

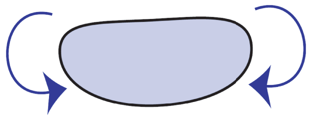


図 2　気泡の周りの流れ

3.　適応的データ構造

気泡による気泡の周りの流体への影響を考慮するためには気泡に対して計算格子を細かくする必要がある．しかしながら，全ての領域を均一に細かくしたのではメモリ使用量が増し，処理に要する時間も大きくなってしまうので，その領域に適した大きさの格子を用いることを考える．非構造格子である適応的データ構造による格子 [3] を用いて，気泡の周りの格子のみを細かくすることとした．

3.1　ツリー型データ構造

適応的データ構造による格子の管理には，ツリー型データ構造を用いた．ツリー型データ構造は流体解析が単純なアルゴリズムによるループ処理により実行でき，セルの探索が簡単な演算によって行える．構造格子と非構造格子のメリットを併せ持っており，流体解析のためのデータ構造に向いているということで，過去の多くの流体シミュレーションに用いられており，それに関する資料も多いということで，本研究においてもこれに習うことにした．

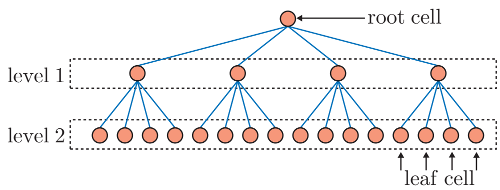


図 3　ツリー構造

ツリー型データ構造では各セルが分岐点となり，各セルを分割することによってツリーが伸びていく．

本研究では，2 次元ではセルを等方的に 4 等分する

“ Quadtree(四分木) ”を，3 次元ではセルを等方的に 8 等分する“ Octree(八分木) ”を用いる．基本になるセルを“ ルートセル (root cell) ”分割されるセルを“ 親セル (parent cell) ”，分割されたセルを“ 子セル (child cell) ”，セルの世代を“ レベル (level) ”，末端のセルを“ リーフセル (leaf cell) ”と呼ぶ [4]．各セルは子セルと親セルへのポインタを持ち，ツリーを辿ることによって各セル間を移動する．

任意の点においてその点を取り囲む最小のセルはリーフセルであるとともに，全ての空間がリーフセルによって覆われているので，数値流体解析はこのリーフセルのみを用いて行う．

3.2 系　　譜

計算を行うためには隣接セルなど，任意の位置のセルを見つける必要がある．しかし，適応的データ構造においては構造格子のように単純に隣接セルを見つけることはできない．よって親子関係を辿って隣接セルを見つけることとなる．この時に，あらかじめ各セルに系譜 [4] を与えておくことで，隣接セルの探索を高速に行うことができるようになる．

系譜とはルートセルから任意のセルまでどのセルを通ってきたかという情報を記憶したもので，セルの親セル内での子セルの位置を x，y 各方向に 0 か 1 で表し，各レベルで得られた 0 か 1 の値を各方向について並べたものを，10 進数に直したものである．図 4 に示すように，系譜はそれ自身と同じ大きさのセルをルートセルに敷き詰めた時のその x，y 座標の値に等しい．すなわち，系譜を用いることによって，今対象とするセルがルートセル内でどこに位置するかというのを簡単に表すことができる．

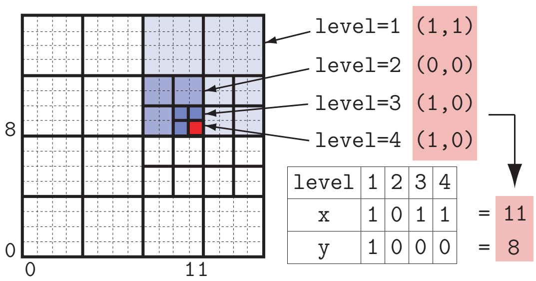


図 4　系譜

親セル内での子セルの位置 x，y から系譜へ，また系譜から親セル内での子セルの位置への変換はシフト演算子を用いて行う．また，図 5 のコードを用いることによって，各方向の系譜が`x,y`，レベルが`level` のセルを見つけ出すことができる．

```
int Pedigree( int cell, int x, int y, int level )
{
    unsigned int tempX = x, tempY = y;
    int tempLevel = "current cell level";
    int child;
    int neighborCell;
    if( "cell has child" ){
        tempX = tempX << ( 32 + tempLevel - level );
        tempY = tempY << ( 32 + tempLevel - level );
        tempX = tempX >> 31;
        tempY = tempY >> 31;
        child = "pointer corresponding ( x, y )";
            neighborCell = Pedigree( child, x, y, level )
    }else{
        neighborCell = cell;
    }
    return neighborCell;
}
```

図 5　セル探索

3.3 セルの分割

ツリー型データ構造ではセルを分割していく必要があるが，分割をするための何らかの情報を基準にしないと適切な計算格子を作成することはできない．そして気泡の周りでセルを細かくする場合，境界に近くなるほどセルが小さくできるような基準が必要である．これらの理由から，セル分割の基準として符号付距離関数とヘビサイド関数 [5] を用いた．

まず，各セルの中心において式 (4) により符号付距離関数 ϕ を定義する．

$$\phi = l - r \tag{4}$$

ϕ は，気泡の中心とセルの中心との距離 l $(l > 0)$ と気泡の半径 r との差で表される．また，符号付距離関数はその符号を調べることによって以下のように気泡の内外の判定にも用いることができる．

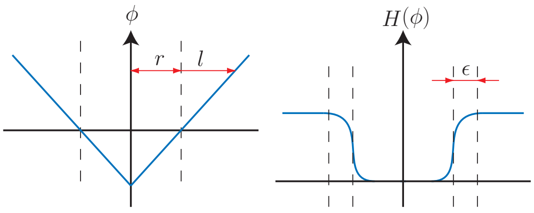
$$\begin{array}{lcl} \phi < 0 & : & \text{気泡内} \\ \phi = 0 & : & \text{気泡の境界上} \\ \phi > 0 & : & \text{気泡外} \end{array}$$

ϕ から式 (5) で定義されるヘビサイド関数の値を計算する．

$$H(\phi) = \begin{cases} 0 & \phi < -\epsilon \\ \dfrac{1}{2}\left(1 + \dfrac{\phi}{\epsilon} + \dfrac{1}{2\pi}\sin\left(\dfrac{\pi\phi}{\epsilon}\right)\right) & -\epsilon \le \phi \le \epsilon \\ 1 & \phi > \epsilon \end{cases} \tag{5}$$

ここで，ϵ は気泡の周りの分割すべきセルの範囲を制御する定数である．

ヘビサイド関数は図 6(b) のように気泡の境界から



(a) 符号付距離関数　　(b) ヘビサイド関数

図 6　セルの分割条件

$\pm\epsilon$ の領域だけ傾きが連続的に変化し，気泡の境界上で傾きが最大となる関数である．気泡の境界付近のセルの中心でのヘビサイド関数の傾きが閾値よりも大きい場合にそのセルを分割していくが，ここで閾値を複数設け，複数回分割させる事によって気泡の境界に近づくにつれてセルを小さくすることができる．本研究においては無条件にすべての領域で均一にレベル 6 までセルを分割し，気泡の付近では最大レベル 8 までセルを分割している．

4.　圧力場と速度場の計算

4.1　異なるセルのレベルでの圧力・速度

適応的データ構造による計算格子においては図 7 の cell1〜cell8 のようにセルの幅が必ずしも隣同士で等しくない．したがって，セルの幅が隣同士で異なっている場合はそのセルの幅での速度・圧力の値に補正する必要がある．

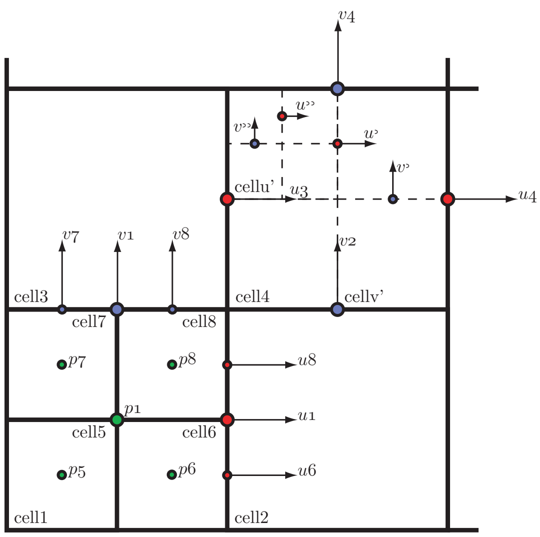


図 7　異なるレベルでの圧力・速度

まず，隣のセルが自分のセルよりも大きいとき，圧力と速度はそのセルの値をそのまま用いる．すなわち，図 7 において cell6 を考える場合，圧力 p_6，x 方向の速度 u_6，そして y 方向の速度 v_6 は

$$p_6 = p_1$$
$$u_6 = u_2$$
$$v_6 = v_3$$

となる．

次に隣のセルが自分のセルよりも小さいとき以下の手順によって圧力・速度を決定する．

圧力は，図 7 において cell1 の圧力 $p1$ の値を cell5〜8 の圧力 $p5$〜8 から求める場合，子セルの圧力の平均値とする (式 (6))．

$$p1 = \frac{p5 + p6 + p7 + p8}{4} \tag{6}$$

速度は，図 7 において $u1$，$v1$ の値を求める場合，cell1 において右側に位置する 2 つの子セル (この場合では cell6 と cell8) の速度の平均値を $u1$ とし，上側に位置する 2 つの子セル (この場合では cell7 と cell8) の速度の平均値を $v1$ とする (式 (7)，(8))．

$$u1 = \frac{u6 + u8}{2} \tag{7}$$

$$v1 = \frac{v7 + v8}{2} \tag{8}$$

計算過程において，図 7 において点線で表された，存在しないセルの位置での速度 u'，v'，u''，及び v'' が必要な場合がある．この場合は仮のセル cellu'，cellv' を考える．u' を求める場合は，cell4 の x 方向の系譜を x，cellu' の x 方向の系譜を x' とし，cell4 と cellu' とのレベルの差を D_l とすると式 (9) となる．v' を求める場合は cell4 の y 方向の系譜を y，cellv' の y 方向の系譜を y' とし，cell4 と cellv' とのレベルの差を D_l とすると式 (10) となる．u''，v'' に関しても同様の方法で求めることができる．

$$u' = u3 - \frac{(x - x'D_l + 1)(u3 - u4)}{D_l} \tag{9}$$

$$v' = v2 - \frac{(y - y'D_l + 1)(v2 - v4)}{D_l} \tag{10}$$

4.2　気泡の速度

流体中の物体は流れによって流され，かつ物体の周りの圧力によっても圧力が低い方向へ流されると考え，気泡の速度は気泡の周りの流体の速度と圧力，及び浮

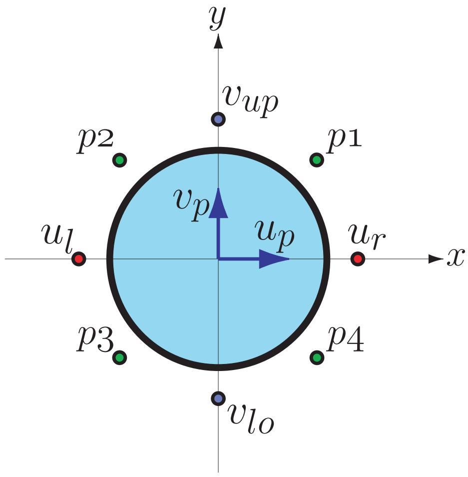


図 8　気泡の速度

力により求める．流体の速度は図 8 のように左右の点と上下の点で別々に考え，圧力は気泡の周りの 4 点を考える．気泡の x 軸方向の速度増分を x 軸方向の速度 u_l，u_r の平均と圧力 $p1$〜$p4$ による速度変化分 ΔU_p の和とし (式 (11)，(13))，y 軸方向の速度増分を y 軸方向の速度 v_{up}，v_{lo} の平均と圧力 $p1$〜$p4$ による速度変化分 ΔV_p，及び浮力 $\mathbf{F}_b$ の和とする (式 (12)，(14))．

$$\Delta U_p = \frac{(p2 - p1 + p3 - p4)}{\sqrt{2}} dt \tag{11}$$

$$\Delta V_p = \frac{(p3 + p4 - p1 - p2)}{\sqrt{2}} dt \tag{12}$$

$$u_p^{n+1} = u_p^n + \frac{u_l + u_r}{2} + \Delta U_p \tag{13}$$

$$v_p^{n+1} = v_p^n + \frac{v_{up} + v_{lo}}{2} + \Delta V_p + \mathbf{F_b} \tag{14}$$

5. 結　　果

図 9(a) は青い円で表される気泡の情報より求めたヘビサイド関数の傾きを表したもので，色が濃くなるほど値は大きくなる．また，気泡の周りのみグリッドを細かく切った結果を図 9(b) に示す．一番大きいセルがレベル 6，一番小さいセルがレベル 8 である．

適応的グリッドを用いたシミュレーションの結果を図 9(c)，図 9(d) に示す．図中の青い円は気泡，赤い矢印は定常的に付加した外力 f を，白線は速度ベクトルを表しており，図が煩雑にならないようにレベルが 6 と 7 のセルの速度ベクトルのみを描画している．

計算時間はレベルが 8 のリーフセルのみを用いて行ったシミュレーションでは 1 ステップ 48.0 秒，図 9(b) のようにレベルが 6〜8 のセルを用いて行った場合は 3.1 秒であった．また使用するメモリ領域は，前者が 13.46MB，後者が 1.06MB であった．これらをふまえると，気泡の数やヘビサイド関数の ϵ の値にも依存するが，適応的に計算格子を設けることによって大幅な計算時間の短縮が図れ，必要なメモリ量も少なくなる．

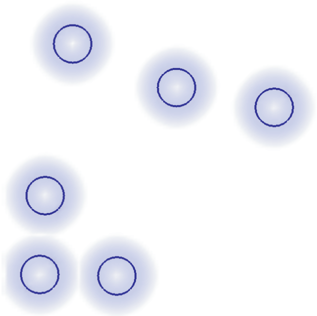
6. ま　と　め

本研究では，リアリティをより高めるために，気泡を考慮した流体シミュレーションを行った．適応的データ構造を用いることにより，処理時間や必要とされるメモリ領域をあまり増加させることなく，気泡周りの流体の流れをシミュレートすることが可能となり，複雑な気泡の動きを再現することができた．

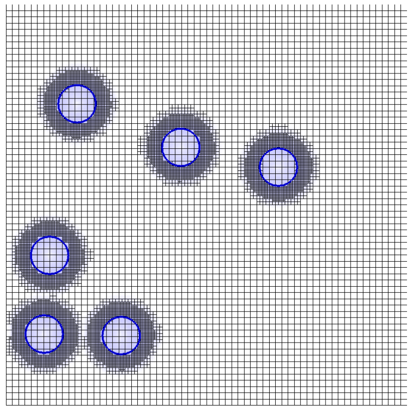
今後の課題としては，カルマン渦などの流体現象を表現できるように数値計算ソルバを改良するとともに，2 次元から 3 次元への拡張，気泡の変形を考慮したアルゴリズムの実装などを行う予定である．

参　考　文　献

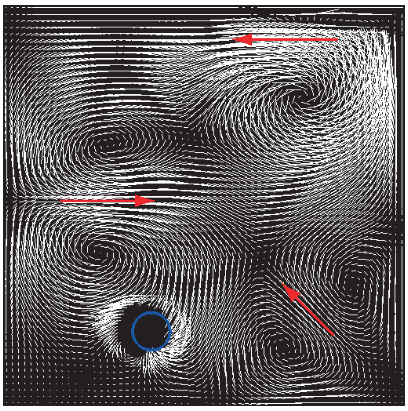
1) Stam, J., " Real-Time Fluid Dynamics for Games, " Proceedings of the Game Developer Conference, 2003.
2) Stam, J., " Stable Fluids, " SIGGRAPH 1999, ACM TOG 26, pp.121-128, 1999.
3) Losasso, F., Gibou, F., Fedkiw, R., " Simulating Water and Smoke with an Octree Data Structure, " SIGGRAPH 2004, ACM TOG 23, pp.457-462, 2004.
4) Ogawa, T., " An Efficient Numerical Algorithm for the Treedata Based Flow Solver, " Computational Fluid Dynamics 2000, pp.337-342, 2000.
5) Osher, S., Fedkiw, R., " Level Set Methods and Dynamic Implicit Surfaces, " Springer Verlag, 2002.



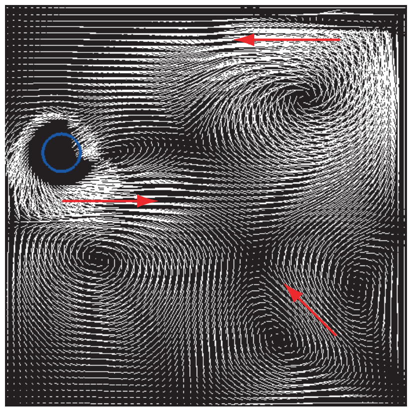
(a) ヘビサイド関数



(b) 適応的グリッド



(c) シミュレーション結果 1



(d) シミュレーション結果 2

図 9 結果