１．緒言

近年、粒子法の一つである Smoothed Particle Hydrodynamics 法(以下SPH法と省略)による流体のシミュレーションが注目されている。SPH法は流体の運動に追従する変数を用いるラグランジュ的な記述に基づく数値計算法であり、オイラー的な記述に基づく格子法に比べると、境界条件が複雑な形状を持つ場合に、境界条件の設定が比較的容易であるという特徴を持っている。SPH法は1977年にMonaghan and Gingold [1], Lucy [2]によって天体物理学の分野で提唱された計算法であり、これまでに数多くの研究がなされている。

またSPH法では流体を運動する粒子群として近似的に記述するために、いわゆる「物理エンジン」との親和性が高い。物理エンジンはシミュレーションの基礎となるシミュレーション環境、オブジェクト生成、物理計算、描画処理のシステムを備えており、プログラマが計算式を直接にプログラムする必要がないという利点があるので、物理エンジンに流体の計算コードを組み込むことは応用上も重要であると考えられる。

本研究では物理エンジン上のSPH法のコードに温度変化の影響を導入し、熱流体の数値シミュレーションコードを実装する方法について検討を行う。

２．流体力学とSPH法の基礎

本章では流体力学の基礎方程式とSPH法による離散化の方法について概観する

２．１　流体力学の基礎方程式

流体を記述するためには質量密度、速度、内部エネルギーの5個の変数が必要であり、これらの時間発展は連続の式、運動方程式およびエネルギー式からなる。

質量密度の発展方程式は連続の式と呼ばれ次の式で与えられる

ここでは質量密度、は質量密度、はナブラ演算子である。また時間微分の演算子は実質微分であり、流体の運動に沿って見た物理量の時間変化を表している。SPH法では密度場は粒子の配置で決まるので、この発展方程式を時間に関して積分することはない。

流体粒子の運動量の発展方程式、すなわち運動方程式はナビエ・ストークス方程式と呼ばれ、次式で与えられる：

ここでは圧力場、は粘性係数、はラプラシアン、は外力の寄与を表している。流体に働く外力の主なものとしては重力、表面張力、電場などの外場の影響、境界との相互作用などが考えられる。

以上の式で物理量を表す従属変数は密度、速度、圧力の5個であるが、発展方程式は密度と速度の4つしかない。圧力の時間変化を決めるための式が必要になる。

流体が理想気体である場合には、圧力は理想気体の状態方程式

によって決められる。上式は単位体積あたりの関係式であり、は単位体積あたりのモル数、は気体定数、は流体の絶対温度である。もし流体の絶対温度が一定であるとすると、上式より圧力と密度は比例関係にあることが分かる。

流体が非圧縮性（）の場合には式(\*,\*)の両辺の発散をとると

という関係式が導かれる。これはいわゆるポワッソン方程式と呼ばれる偏微分方程式であり、これを積分して圧力場が得られる。

流体が非圧縮性を仮定できる場合に、流体の運動に運度変化の寄与を導入する方法としてブシネスク近似がある。ブシネスク近似とは温度の流体の質量密度を基準状態の温度, 密度からの差と熱膨張係数を用いて次のように近似をする (文献 [3]、Chapt.1)：

ここで温度の基準温度との差は絶対温度に比べて小さく、密度変化の寄与は浮力項以外には効かないものと仮定する。温度の時間変化は温度の基準状態からのずれが時間とともに拡散して、基準状態の温度に緩和すると仮定し、拡散方程式

を用いることが多い。

２．２　SPH法による離散化

これらは連続な場であるが数値シミュレーションにおいては、場を有限個の点上のデータとして離散化して近似をする必要がある。この離散化の方法にはオイラー的な離散化と ラグランジュ 的な離散化の2種の方法がある。一般的な差分法、有限要素法による数値解法は前者に相当し、本研究で取り扱う SPH 法は後者にあたる。ラグランジュ的な手法の利点としては、空間上に計算格子点をメッシュ状に設定する必要がないので、Euler 的な手法と比較して、複雑な形状の境界を持つ問題を解くのに適していることが挙げられる。

SPH法による離散化の手続きの考え方は次のようになる [4]。

まず一般的に連続的な場（とする）はディラックのデルタ関数を用いて次のように書くことができる：

SPH法ではまずデルタ関数の部分を平滑化核関数（スムージング・カーネル）と呼ばれる関数（とする）を用いて場を次のように近似した場（とする）を考える：

ここではのウィンドウ幅である。平滑化核関数は次の特徴をもつ。正の値を持つ：空間積分すると1になるよう規格化されている：ウィンドウ幅の極限でディラックのデルタ関数に漸近する：

一般に核関数としては原点に関して球対称な関数をとる：。続いて、被積分関数のを連続場ではなく*N*個の離散点 における値を用いて

と近似をする、ここでは各離散点上で与えた量のデータ、は各離散点で定義された体積要素であり、積分のを近似したものである。ここでさらに質量、体積、質量密度の間に成り立つ関係式を代入して

と近似をする。この式が場のSPH近似の式となる。ここで注意すべき点は次の二つである。まず離散点上で与えたデータは場の近似値とは異なること：

またとは独立ではなく、にを代入することにより得られる次の関係式を満たすこと：

実際の計算では質量を固定し、質量密度を計算の各ステップで再計算をする。

　次にSPHにおける場の量の空間変数に関する導関数だが、SPH近似の式(\*,\*)をに関して微分して

を得る。この式から分かるようにに関する微分はカーネル関数のところだけにかかる。他の空間変数に関する導関数も同様であるからSPH法における場の勾配はカーネル関数の導関数を用いて次のように書けることがわかる：

これより番目の粒子の位置における勾配場の値は次のようになる：

SPH法での導関数の計算法として恒等式

を用いて添字の入れ替えに対して反対称な定式化を考えることができる：

またその一方で恒等式

を用いて添字の入れ替えに対して対称な定式化も考えることができる：

これらの添え字に関して対称または反対称な定式化をすることによって、計算精度と数値計算的な安定性が期待できる。

２．３　時間に関する数値積分(文献 [5])

流体の流れのシミュレｰションを行うために、グローバルな固定時間ステップを用いて、各々の粒子を進める必要がある。 (4.2)式は粒子加速度を計算するために用いられ、新しい粒子位置は数値的に加速度を積分することで得られる。

(1)オイラー法・陰的オイラー法

オイラー法(4.60)式は一般的な方法であり、現在の速度を用いて次の位置を計算し、次の速度を計算する。

それに対して、陰的オイラー法(4.62)式は次の位置の計算に速度を更新する式

から得られる次の時間の速度を使う。

(2)リープ・フロッグ法

リープ・フロッグ法は2次精度の時間積分スキームであり、SPH法の積分スキームとしてよく用いられている方法である。半ステップ前の速度と半ステップ先の速度の二つの速度を用いて1ステップ先の位置を求めるという特徴がある。

このスキームでは速度の積分時刻は速度の積分時刻とだけずれるので、位置の時間積分値と同じ時刻での速度データを求めるには式

を用いる必要がある。

３．物理エンジンの概要

物理エンジンとは運動方程式に基づいて物体の運動を自動的に計算するための application programing interface (以下APIと省略)のセットのことである。このAPIを用いることでプログラマは物体の衝突や摩擦、運動方程式などの煩わしいプログラミングをする必要がなく、物体の形状、初期条件、地形などの境界条件を入力することによりシミュレーションを実行できる。

現在、複数の物理エンジンが公開されており、主なものとして Bullet Physics （以下BPと省略） [6]、PhysX [7], Havok [8], ODE [9]がある。これら物理エンジンによる物体の運動等の力学現象の再現性のベンチマークが Boeing and Bräunl によって行われ、全体としてBPが安定した結果を出すことが分かった [10]。

さらにBPはZlibライセンス [11]の下で提供されているオープンソースなソフトウェアライブラリであり、改変が容易であるという特徴がある。したがって本研究では熱流体コード実装のためのプラットフォームとしてBPを採用することにした。

本研究でもちいたBPのバージョンは 2.81 rev2613 である [12]。このパッケージには流体オブジェクトとして Hoetzlein による SPH 法の実装である Fluids v.2が標準的に添付して配布されている [13]。Table ? にBPの主な諸元をまとめた。

BPでは時間積分のスキームとして位置、速度に関してリープ・フロッグ法が用いられている。

本研究ではこの実装に対し、Lee がさらに改善を加え、BPのネイティブなオブジェクトとSPH法の流体粒子の相互作用が実装されたコードである FluidDemo をベースにして研究をすすめる [14]。

BPのソースコードはC++言語を用いて書かれており、本研究では開発、コンパイル、実行のプラットフォームとして Visual Studio 2008 を用いた。

Table. 3.1: principal features of Bullet Physics Library.

|  |  |
| --- | --- |
| プラットフォーム | Win32, Mac, Linux, PS3, XBOX360 |
| 実装されている主な材質特性 | 反発, 静止摩擦 |
| サポートしているジオメトリ  (物体形状) | Capsule, Box, Sphere, Plane, Cone, Convex Mesh, Compound Object, Heightfield , Triangle Mesh |

SPHコードの構成

本研究で利用した数値計算コード FluidDemo の流体の運動を計算する部分のコードの依存関係をFigure ? にまとめた。図において各ボックス内の1行目にはソースコードのファイル名を2行目にはクラス名、関数名を、最後に動作についてのコメントを記載してある。

時間積分のコードは(1)粒子の配置が変わったことによる密度、圧力場の再計算、(2)得られた圧力と速度場を用いての圧力勾配、粘性力の計算、(3)重力などの外力場の計算、(4)リープフロッグ法による積分という4ステップに分けて、実行されている。

btThermalSolver.h

updateTemperature()

温度のEuler積分の反映

btThermalSolver.h

computeTemperature()

温度拡散の計算

FluidSolver.cpp

FluidSolver::integrate()

積分コードのラッパー

btFluidSolver.cpp

integrateParticle()

重力の計算

位置、速度のLeapFrog積分

btFluidSolver.cpp

computeForceNeighborTableSymmetric()

粒子毎の圧力勾配、粘性力の和の計算

btFluidSolver.h

class btFluidSolverSph : public btFluidSolver

virtual void stepSimulation()

FluidDemo.h

FluidDemo::clientMoveAndDisplay()

btFluidSolver.cpp

FluidSolver::sphComputePressure()

圧力m\_pressure[], 密度 m\_density[]の更新

btFluidSolver.cpp

FluidSolver::sphComputeForce()

他成分との粒子毎の圧力勾配、粘性力の和の計算

Figure 3.1: dependency diagram of functions in the simulation code

４．温度の影響の導入

本研究では温度変化に伴う質量密度の変化の影響をブシネスク近似を用いて取り込むことにする(文献 [15]、Chapt.1)。ブシネスク近似されたなナヴィエ・ストークス方程式は次の式で与えられる：

ここでは実質微分、は速度場、は質量密度場、は圧力場、は重力加速度ベクトル、は流体の熱膨張係数、は流体の平均温度からのずれを表す場(以下、温度ずれ場と呼ぶ)、は動粘性係数である。温度ずれ場の支配方程式は次のようになる：

ここでは温度拡散係数である。

この支配方程式の離散化に際して、本研究では拡散項を粘性項との類比で求めることにした。まず本研究で取り扱ったコードでは流体の粘性項のモデルとしてラプラシアンを次式で近似したモデルを用いている [5]：

ここで添字はその物理量を持つ流体粒子の番号を表し、である。本研究ではこの対称化されたラプラシアンの表現を温度に置き換えた式

を温度の発展方程式の拡散項に用いた。温度の積分のスキームはオイラー法とした。

　具体的なコードの変更点としては、(1)温度のデータを記録するための変数を流体粒子の構造体の定義に加え（btFluidParticles.hのbtFluidParticle構造体の変更）、(2)拡散方程式の拡散項の計算を粘性力の計算の後に加え（btFluidSolver.cppのcomputeNeighborTableSymmetric()関数から、拡散項を計算するbtThermalSolver.hのcomputeTemperature()関数を呼び出すように追加）、(3)積分コードにオイラー法による温度場の積分を加えた（btFluidSolver.cppのintegrateParticle()関数から、温度場の積分コードであるbtThermalSolver.hのupdateTemperature()を呼び出すように追加）。追加した関数の呼出し関係については Figure 3.1 に記している。

５．熱流体シミュレーションコードのテスト

本章では本研究で作製したシミュレーションコードのテストの概要を述べる。以下、長さ、質量、時間、温度の数値はシミュレーションコード内で用いた値をそのまま単位を付けずに表す。物体の次元の数値はx×y×zの順であり、yが鉛直方向に割り当てられている。コードでは各数値はメートル、キログラム、秒、ケルビンを想定して書かれているが、各数値はスケールを変更することによって、さまざまな単位に換算して考えることができるので、以下では数値は単位をつけずにシミュレーションで用いた値をそのまま表記することにする。

テスト領域は30×30×30のサイズの立方体の領域を準備し、その底面中央に10×1×10の「台」(CollisionShapeオブジェクト)の上に、20×16×1の直方体(CompoundShapeオブジェクト)を2枚、十字型に組んだ「パドル」をHinge2Constraintオブジェクトで結合し、「攪拌機」を設置した(Figure 5.1参照)。攪拌機は回転角速度(rad/s)で回転させ、流体を撹拌し、熱伝達のふるまいを変えることが期待される。

この領域内に20000個の粒子を配置し、初期値問題を積分する。積分の時間刻みは*t*=0.003で実行した。*t*の積分への影響を見るために*t*=0.001のランも実行した。

計算コードにおける主要な固定パラメーターを Table 5.1 に、テストのためのシミュレーションパラメーターを Table 5.2 に掲載する。

初期条件において温度Tの初期条件は、粒子全体の温度の平均が0となるようにとった。温度の初期条件は温度をT=0で均一に設定したもの、正と負のずれを設定したもの2種の計3種類の設定をし、互いに比較することにした。正負の温度を設定する初期条件においては、初期に粒子は z > 15 の領域に T > 0 となる粒子を、z < 15 の領域に T < 0 となる粒子を配置した。この配置は力学的に不安定であり、冷たい流体は下に、暖かい流体は上に移動しようとする。そのためシミュレーション開始直後に流体は激しく運動する。

本テストではシミュレーションの可視化は、各流体粒子に温度を表現する色を付け、粒子の運動と温度の変化を追跡できるようにした。

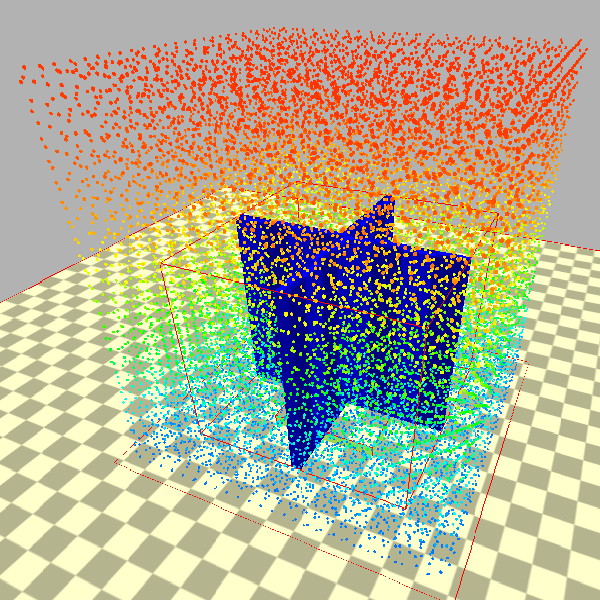


Figure 5.1: configuration of thermal fluid flow simulation.

本研究では、この配置に関し、次の点に着目してシミュレーションを行った：(1)静止状態の流体における熱伝導の効果(A-C-E)、(2)等温の流体において撹拌が圧力分布、粒子数分布に与える効果(A-B)、(3)熱流体における撹拌が熱伝導に与える影響(E-H-F)、(4)撹拌された流体における初期温度分布の影響(B-D-F)、(5)攪拌機のサイズの熱伝達に与える影響(F-G)、(6)熱伝導係数の静止流体の熱伝達に対する影響(E-I)、(7)熱伝導係数の撹拌流体の熱伝達に対する影響(F-J)、(8)積分時間刻みの静止流体の熱伝達に対する影響(E-K)、(9)積分時間刻みの撹拌流体の熱伝達に対する影響(F-L)。カッコ内のアルファベットはシミュレーションの名前である(Table 5.2 参照)。各シミュレーションの結果、得られた温度、圧力、粒子数の深さ方向の分布のグラフを Figure 5.2-5.13 に示す。グラフの値はシミュレーション領域を深さ方向に10個に等分割し、それぞれのブロックでの総和や平均値を示している。以下、これらの結果について考察を行う。グラフには積分のステップ数が1,000ステップ毎のデータを1,000ステップ目から10,000ステップまで表示している。

(1) 静止状態の流体における熱伝導の効果(A-C-E)

このシミュレーションはSPH法のコードにおける、流体の静水圧平衡の状態、ならびに流体の運動がない場合の熱伝導の時間発展をチェックすることにある。

まず10本のグラフがほぼ一つに揃っている。これは流体の運動が1000ステップ以内に緩和して、静水圧平衡の状態に緩和していることを示している。

圧力のグラフより分かるように、圧力は流体の上面で０になっている。これはこのシミュレーションの設定において、流体の上に物質がなんら設定されていないので、圧力ゼロで流体が動かない状態になっていることを示している。圧力は深さとともに増えているが、深さに比例しているわけではない。力学的平衡状態の流体のつり合いの式はナビエ・ストークス方程式より

で与えられる。シミュレーションの設定では重力加速度は一定なので、密度が深さによって変化していることを示している。粒子数のグラフを見ると、粒子数も深さとともに増えていることが分かる。シミュレーションの最上位の層と最深層で粒子数に5倍程度の違いが見られる。すなわちSPH法のスキームに基づく圧縮性が反映されている。

(2) 等温の流体において撹拌が圧力分布、粒子数分布に与える効果(A-B)

(3) 熱流体における撹拌が熱伝導に与える影響(E-H-F)

(4) 撹拌された流体における初期温度分布の影響(B-D-F)

(5) 攪拌機のサイズの熱伝達に与える影響(F-G)

(6) 熱伝導係数の静止流体の熱伝達に対する影響(E-I)

(7) 熱伝導係数の撹拌流体の熱伝達に対する影響(F-J)

(8) 積分時間刻みの静止流体の熱伝達に対する影響(E-K)

(9) 積分時間刻みの撹拌流体の熱伝達に対する影響(F-L)

粒子数、圧力の分布はさまざまな要素を変えて計算したすべての結果において、ほとんど値に差がなかった。これは圧力、粒子数の値は鉛直方向の静水圧平衡で値が決まっていることを表している。

Table 5.1: list of principal parameters in the simulation code

|  |  |  |
| --- | --- | --- |
| ファイル名 | btFluidParameters.h |  |
| 構造体名 | struct btFluidParametersGlobal |  |
| m\_timeStep | 時間積分の刻み幅(s) | 0.003, 0.001 |
| m\_simulationScale |  | 0.004 |
| m\_speedLimit | 加速度/力の限度(m/s) | 200 |
| m\_sphSmoothRadius | SPH粒子の相互作用半径(m) | 0.01 |
| m\_sphRadiusSquared | m\_sphSmoothRadius2 |  |
| m\_poly6KernCoeff | 密度計算のためのpoly6kernelの係数 |  |
| m\_spikyKernGradCoeff | 圧力計算のためのspikyKernelの勾配係数 |  |
| m\_viscosityKernLapCoeff | 粘性力計算のための粘性カーネルのラプラシアンの係数 |  |
| 構造体名 | struct btFluidParametersLocal |  |
| m\_volumeMin | 粒子の移動範囲の下側の境界(m) | (-15,-10,-15) |
| m\_volumeMax | 粒子の移動範囲の上側の境界(m) | (15,30,15) |
| m\_gravity | 重力(m/s2) | (0,-9.8,0) |
| m\_viscosity | 値が高いほど粘性が高くなる(Pa・s) | 1.5 |
| m\_restDensity | 静止密度(kg/m3) | 1000 |
| m\_particleMass | 粒子の質量(kg) | 0.00020543 |
| m\_stiffness | 気体定数で値が高いほど圧縮が少なく不安定になる(J) | 1 |
| m\_particleRadius | 粒子の衝突判定半径(m) | 1 |
| m\_boundaryStiff | 弾性係数、境界の反発力の大きさ | 20000 |
| m\_boundaryDamp | 減衰係数、境界の反発力における相対速度への影響 | 256 |
| m\_boundaryFriction | 境界摩擦、接線の速度を減少させ、高い値だと不安定になる | 0.00075 |
| m\_particleDist | btFluidEmitterの粒子間隔を決定する(m) |  |

Table 5.2: principal parameters of simulation runs.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Run |  |  | *T*(0) |  | *t* |  |  |  | note |
| A | 0 |  | 0 |  | 0.003 |  | 0.00001 |  |  |
| B | 1000 |  | 0 |  | 0.003 |  | 0.00001 |  |  |
| C | 0 |  | -25,25 |  | 0.003 |  | 0.00001 |  |  |
| D | 1000 |  | -25,25 |  | 0.003 |  | 0.00001 |  |  |
| E | 0 |  | -50,50 |  | 0.003 |  | 0.00001 |  |  |
| F | 1000 |  | -50,50 |  | 0.003 |  | 0.00001 |  |  |
| G | 1000 |  | -50,50 |  | 0.003 |  | 0.00001 |  | small paddle |
| H | 500 |  | -50,50 |  | 0.003 |  | 0.00001 |  |  |
| I | 0 |  | -50,50 |  | 0.003 |  | 0.0001 |  |  |
| J | 1000 |  | -50,50 |  | 0.003 |  | 0.0001 |  |  |
| K | 0 |  | -50,50 |  | 0.001 |  | 0.00001 |  |  |
| L | 1000 |  | -50,50 |  | 0.001 |  | 0.00001 |  |  |

Figure.5.2: static mechanical equilibrium state without thermal effect. Time development of vertical distributions of pressure(left) and particle number(right): case A, =0, *T*(0)=0, *t*=0.003, =0.00001

Figure.5.3: effect of stirring on the distribution. Time development of vertical distributions of pressure(left) and particle number(right): case B, =1000, *T*(0)=0, *t*=0.003, =0.00001

Figure.5.4: effect of thermal conduction. Time development of vertical distributions of temperature(left), pressure(middle) and particle number(right): case C, =0, *T*(0)=25, 25, *t*=0.003, =0.00001

Figure.5.5: effect of stirring on thermal conduction. Time development of vertical distributions of temperature(left), pressure(middle) and particle number(right): case D, =1000, *T*(0)=25, 25, *t*=0.003, =0.00001

Figure.5.6: effect of thermal conduction. Time development of vertical distributions of temperature(left),

pressure(middle) and particle number(right): case E, =0, *T*(0)=50, 50, *t*=0.003, =0.00001

Figure.5.7: effect of stirring on thermal conduction. Time development of vertical distributions of temperature(left), pressure(middle) and particle number(right): case F, =1000, *T*(0)=50, 50, *t*=0.003, =0.00001

Figure.5.8: effect of paddle size on the stirring. Size of each paddle is reduced to 10x16x1. Time development of vertical distributions of temperature(left), pressure(middle) and particle number(right): case G, =1000, *T*(0)=50, 50, *t*=0.003, =0.00001

Figure.5.9: effect of slow stirring on thermal conduction. Time development of vertical distributions of temperature(left), pressure(middle) and particle number(right): case H, =500, *T*(0)=50, 50, *t*=0.003, =0.00001

Figure.5.10: effect of thermal conduction coefficient on thermal conduction. Time development of vertical distributions of temperature(left), pressure(middle) and particle number(right): case I, =0, *T*(0)=50, 50, *t*=0.003, =0.0001

Figure.5.11: effect of thermal conduction coefficient on thermal conduction with stirring. Time development of vertical distributions of temperature(left), pressure(middle) and particle number(right): case J, =1000, *T*(0)=50, 50, *t*=0.003, =0.0001

Figure.5.12: effect of integration time step on thermal conduction. Time development of vertical distributions of temperature(left), pressure(middle) and particle number(right): case K, =0, *T*(0)=50, 50, *t*=0.001, =0.00001

Figure.5.13: effect of integration time step on thermal conduction with stirring. Time development of vertical distributions of temperature(left), pressure(middle) and particle number(right): case L, =1000, *T*(0)=50, 50, *t*=0.001, =0.00001

６．結言

本研究では物理エンジン Bullet Physics 上の SPH 法の実装の一つである FluidDemo をベースにして、流体の運動に対して温度変化に伴う浮力の影響を導入することを試みた。シミュレーションコードに対し温度変数の導入と温度拡散による時間変化、ならびにブシネスク近似を用いた浮力を実装し、シミュレーションを行った。その結果、温度変化に伴う流体の挙動の変化を実現することができた。

残された課題として、シミュレーション領域の境界条件となるBPの剛体オブジェクトに関して、温度のパラメーターを持たせ、流体と境界との接触による熱伝達、その結果としての温度変化による流体の挙動への影響を組み込むことがある。このためには粒子の近傍の剛体オブジェクトの探索が不可欠である。BPにはこの探索のための関数としてRaycastが実装されており、これを熱伝達の計算に応用することができると考えられる。

またSPH法では流体の質量密度は流体の配置によって変化するので、これはシミュレートする流体に圧縮性があることに相当している。これに対し現実的で身近な流体の運動では非圧縮性が仮定できるものが多い（文献 [16], §3.6）。そこで非圧縮性をなるべく実現できるスキームが求められる。SPH法に非圧縮性を導入する研究は、例えば予測子・修正子法と組み合わせたものが、Solenthaler and Pajarola によってなされている [17]。このような非圧縮性をどのようにコードに組み込むかを検討する必要がある。

参考文献

1. Monaghan, R.A. Gingold and J.J.,. Mon. Not. R. Astron. Soc., Vol 181, pp. 375–89, 1977..

2. L.B. Lucy,. Astron. J., Vol 82, pp. 1013–1024, 1977..

3. Chandrasekhar, S., "Hydrodynamic and hydromagnetic stability",. Oxford Univ. Press, 1961.

4. Monaghan, J. J.,. Ann. Rev. Astron. Astrophys., Vol. 30 (A93-25826 09-90), p. 543-574, 1992.

5. Kelagar, M., "Lagrangian Fluid Dynamics Using Smoothed Particle Hydrodynamics",. University of Copenhagen, 2006. http://image.diku.dk/projects/media/kelager.06.pdf.

6. bulletphysics.org, "Game Physics Simulation". http://bulletphysics.org/wordpress/.

7. nVIDIA, PhysX,. http://www.nvidia.co.jp/object/physx\_new\_jp.html.

8. Havok.com Inc, Havok,. http://www.havok.com/.

9. Russell Smith, Open Dynamics Engine,. http://www.ode.org/.

10. Boeing, A., and Braeunl, T., proceedings of GRAPHITE '07, doi:10.1145/1321261.1321312.

11. Open Source Initiative, "The zlib/libpng License",. http://opensource.org/licenses/zlib-license.php.

12. bulletphysics.org, "Game Physics Simulation", http://bulletphysics.org/wordpress/.

13. Hoetzlein, R. C., www.rchoetzlein.com, "engineering",. http://www.rchoetzlein.com/eng/index.htm.

14. rtrius, Bullet-FLUIDS, GitHub,. https://github.com/rtrius/Bullet-FLUIDS. 本文では著者名として主なソースコード内の署名である Jackson Lee を用いた。.

15. Chandrasekhar, S., "Hydrodynamic and hydromagnetic stability",. Oxford Univ. Press, 1961.

16. Batchelor, G. K., "An introduction to fluid dynamics",. Cambridge Univ. Press, 1967.

17. Solenthaler B., and Pajarola R.,. ACM Transactions on Graphics, Vol. 28, No. 3, Article 40, 2009.

18. Batchelor, G. K., "An introduction to fluid dynamics", Cambridge Univ. Press, 1967.