粒子衝突シミュレータの制作とその応用

６年Ｂ組　西田　惇
指導教諭　米田　隆恒

１．要約

「熱力学」や「気体の状態変化」で学習する公式や事象は、理屈は理解できるが視覚的に確認したことがないものがほとんどである。そこで CPP 言語を用いて粒子衝突シミュレータを制作し、力学的エネルギーの保存、気体分子運動論、ブラウン運動、 Maxwell の速度分布、エントロピー増大の法則を視覚的に確認できるソフトウェアを制作した。今回は制作期間が７日間と非常に短かったため、多くの公式を可視化することはできなかったが、制作した本ソフトウェアを用いることによって、より感覚的に各公式・事象を認識できるようになった。

キーワード　粒子衝突、熱力学、シミュレータ、C++、並列処理

２．研究背景

中等教育課程で学習する熱力学に関する公式・事象について視覚的に確認した例はほとんどない。そこで、より直感的に事象を理解するため、粒子衝突シミュレータを制作し、いくつかの公式・事象についてコンピュータ上で再現し、視覚化した。

３．目的

粒子衝突シミュレータを制作し、さらに、熱力学及び気体に関するいくつかの公式・事象をコンピュータ上でシミュレートできるソフトウェアを制作する。また、粒子衝突に関する計算時間を短縮する手法について考察する。

４．研究内容

（１）仮説

<研究Ⅰ>「粒子衝突シミュレータの制作」

力積の定義と反発係数の公式等を用いて粒子が衝突する様子をシミュレートできる。

<研究Ⅱ>「数理モデルの実装」

制作したシミュレータでいくつかの公式・事象を再現できる。

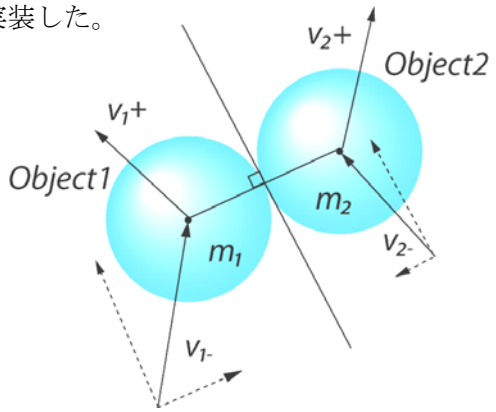
<研究Ⅲ>「並列処理プログラミングの導入と検討」

並列処理プログラミングを導入することにより処理速度を向上させることができる。

（２）研究方法

<研究Ⅰ>「粒子衝突シミュレータの制作」

CPP 言語を用いて Windows Form Application を制作し、粒子衝突前の速度から衝突後の速度を求めるアルゴリズムを実装した。



$$J = m_1(v_{1+} - v_{1-})$$
$$-J = m_2(v_{2+} - v_{2-})$$
$$e = |v_{1+} - v_{2+}| / |v_{1-} - v_{2-}|$$

J：力積 m：質量 v：速度 －：衝突前 ＋：衝突後

上記の３式から次式を求めることができる。

$$J = -Vr(e+1)/(1/m_1 + 1/m_2)$$

ただし、$Vr = v_{1-} - v_{2-}$

各粒子に作用する力積がわかれば、質量で割ることにより衝突後の速度を求めることができる。

$$v_{1+} = v_{1-} + J/m_1 \qquad v_{2+} = v_{2-} + (-J/m_2)$$

<研究Ⅱ>「数理モデルの実装」

今回は以下の公式・事象について視覚化を行った。

① 力学的エネルギー保存則
② 気体分子運動論(平均運動エネルギー)
③ ブラウン運動
④ Maxwell の速度分布
⑤ エントロピー増大の法則

<研究Ⅲ>「並列処理プログラミングの導入と検討」

搭載されたデュアルコア CPU の性能を最大限引き出すために OpenMP を用いて並列処理プログラミングを導入する。並列化した場合としない場合の処理時間を計測し、その効果について考察する。

本研究の仮説を検証するソフトウェア開発環境は以下の通りである。

OS : Windows XP HE SP3
Hardware : Intel Core2Duo P8400,2GB
IDE : VisualStudio2008 PE SP1

(1)研究結果

<研究Ⅰ>「粒子衝突シミュレータの制作」

制作した粒子衝突シミュレータを以下に示す。

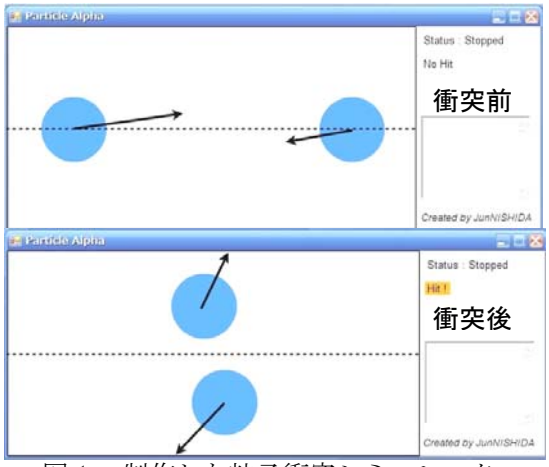


図１　制作した粒子衝突シミュレータ

<研究Ⅱ>「数理モデルの実装」

制作した粒子シミュレータに各数理モデルを実装した。フォームには、初期値設定コントロール(粒子数、質量、半径、反発係数、温度、圧力)、各パラメータを表示するグラフを配置した。

図２　数理モデルを実装したシミュレータ

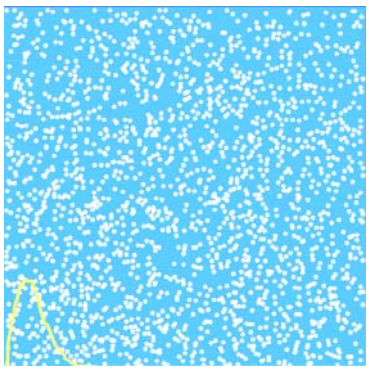


図３　粒子の初期配置状態。
初期座標・初期速度は乱数で決定する。
下部の黄色のグラフは粒子の速度分布を示す。

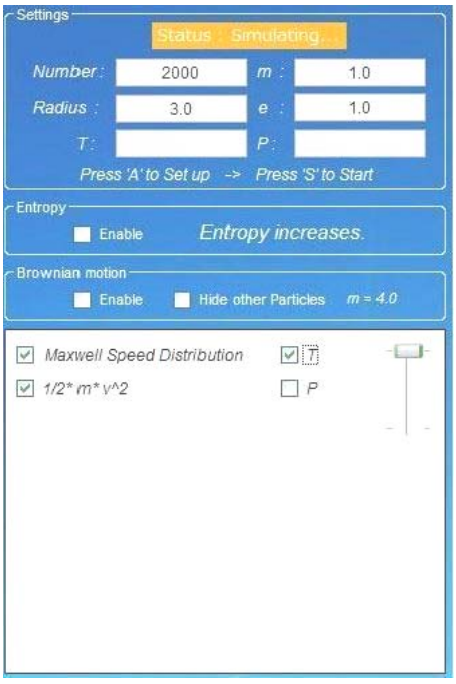


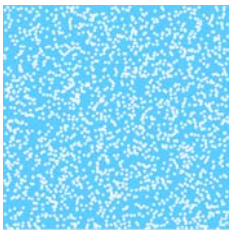
図４　初期値設定部及びグラフ表示部

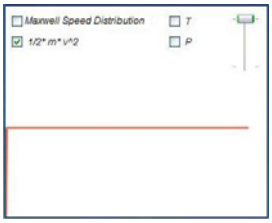
各公式・事象のシミュレーション結果を次に示す。

① 力学的エネルギー保存則の視覚化

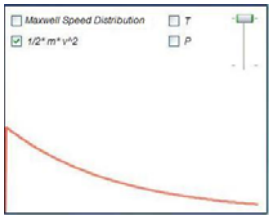
反発係数を変化させると、エネルギーが変化する様子がわかる。

$$E = \sum \frac{1}{2} m v^2$$

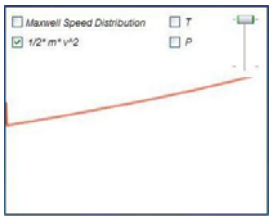




e =1.00



e =0.99



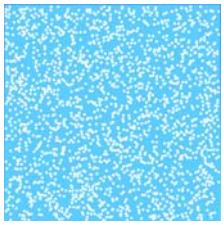
e =1.002

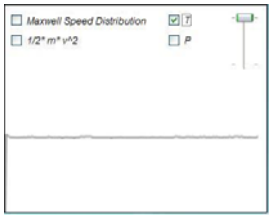
配置した粒子 (粒子数 2000)

② 気体分子運動論(平均運動エネルギーは絶対温度に比例)の視覚化

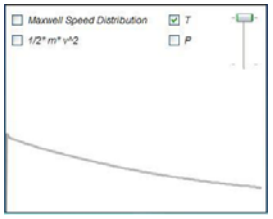
粒子の速度を変化させると、温度が変化する様子がわかる。

$$\frac{1}{2} m \overline{v^2} = \frac{2}{2} kT$$

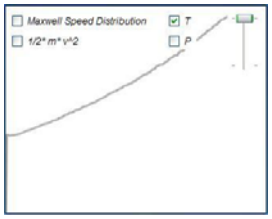




e =1.00



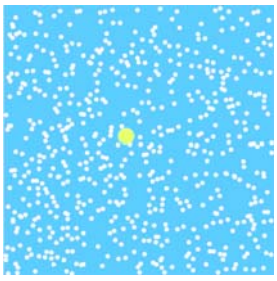
e =0.99



e =1.01

配置した粒子 (粒子数 2000)

③ ブラウン運動の視覚化



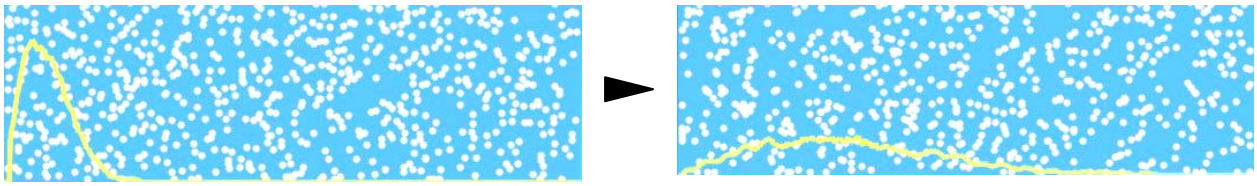
粒子の中にコロイド粒子を配置した様子

コロイド粒子の軌跡のみを表示した様子

(粒子数 2000 、コロイド粒子の質量は粒子の 5 倍)

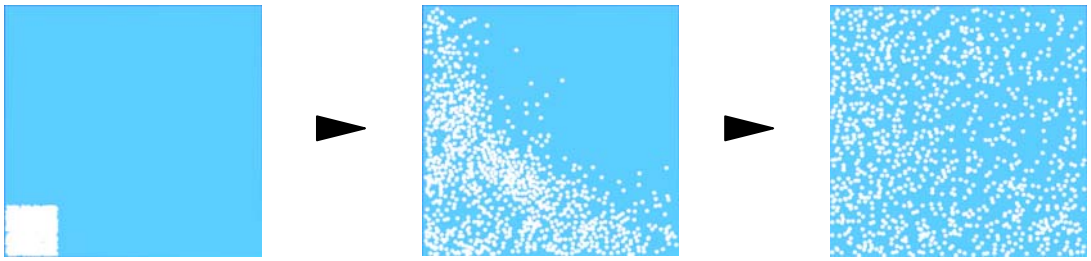
④ Maxwell の速度分布の視覚化

温度を次第に上げていくと、速度分布がシフトする様子がわかる。



粒子の速度が全体的に遅い時　　粒子の速度が全体的に速い時

⑤ エントロピー増大の法則の視覚化



<研究Ⅲ>「並列処理プログラミングの導入と検討」

デュアルコア CPU をコンピュータが搭載していても、ソフトウェア側が並列処理に対応していなければ２個あるコアのうち大半の処理が片方で行われてしまい、マシンのパワーを最大限に引き出せない。

そこで、For 文中に並列処理用 API「OpenMP」を導入し、衝突計算を２つのコアに分散させ、並列処理させた。

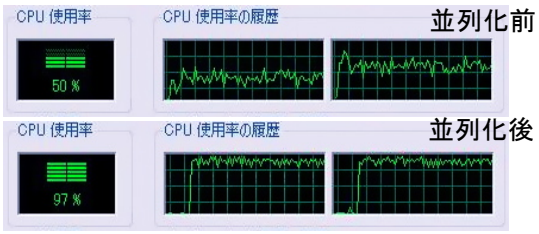


図５　並列処理プログラミング導入の効果
（上：並列化前　下：並列化後）

図５は並列化前と並列化後の CPU 使用率のグラフを示す。並列化前は両方のコアの性能が十分に生かされていないが、並列化後は、両コア共に CPU 使用率がほぼ100%になっている。

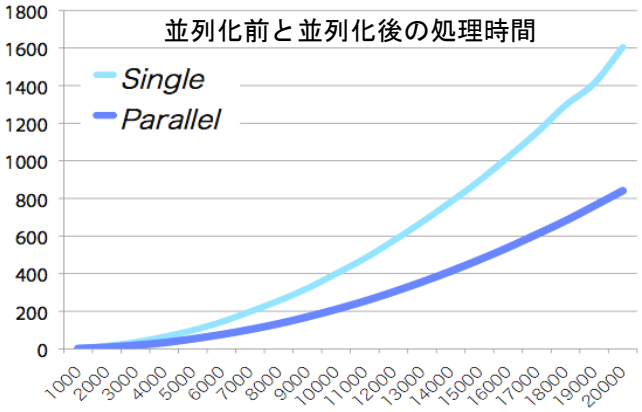


図６　並列処理プログラミング導入の効果
縦軸は処理時間(ms)，横軸は粒子数
Single：並列化前　Parallel：並列化後

図６は並列化前と後の処理時間の比較を表したものである。縦軸は処理時間(ms)、横軸は粒子数を示し、粒子数 1000 ～２万の時の処理時間をそれぞれ計測した。グラフより、衝突計算を各コアに分散して並列処理することにより処理速度が最大２倍高速になることがわかった。

５．考察

並列処理について、搭載するコア数を増やすことによって処理速度がさらに高速になると推測される。さらに、衝突計算だけでなくモーションキャプチャや信号処理等、For 制御文を多用する他のプログラムにおいても高速化が期待できる。実際の授業で使用できるようにするには、UI の改善が必要であると考えられる。

６．まとめと今後の展開

（１）まとめ

粒子を配置した領域内で粒子の衝突、物理・化学公式がシミュレート可能であることがわかった。さらに並列処理プログラミングを導入することにより処理速度を飛躍的に向上できることが検証された。これらにより、より感覚的に各公式を理解できるソフトウェアを提示できた。

（２）今後の展開

より忠実に分子の振る舞いを再現するため、分子間力をパラメータに追加したい。これにより、物体の三態の変化、熱の伝導、熱平衡、対流、放射などのシミュレートをより忠実に再現できるようになる。さらに、シミュレート領域を三次元へ拡張し、現実空間との差異を縮めていきたい。処理速度について、汎用計算を GPU に任せる「GPGPU」、描画処理の「ハードウェアアクセラレーション」を実装することによりさらに処理速度が向上すると考えられる。最終的には、中等教育課程での実際の授業で使用できるように UI の整備、さらなる機能の追加を行っていきたい。

７．参考文献

[1] 「リードα　物理Ⅰ・Ⅱ」，数研出版
[2] 「リードα　化学Ⅰ・Ⅱ」，数研出版
[3] 「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
[4] VisualC++2005，SoftBank Creative
[5] MSDN ライブラリ，Microsoft

８．謝辞

顧問の米田隆恒先生にご指導をいただきました。ありがとうございました。