# Algorithm

擬似コードは、以下の通り。

**STEP1 （密度の更新)**1 for each particle   
2   
2 for each neighbor   
3   
4 if   
5 (A)

**STEP2（容器による密度の更新）**  
1 for each particle   
2 for each 容器のparticle   
3   
4 if   
5 (A)

**STEP3（力の更新）**1 for each particle   
2 ,   
3 for each neighbor   
4   
5 if   
6 (B)  
7 (C)  
8

**STEP4（容器による力の更新）**  
1 for each particle   
2 ,   
3 for each 容器のparticle   
4   
5 if   
6 (B)  
7 (C)  
8

**STEP5（移動）**  
1 for each particle   
2   
3   
4   
5 もし容器の外にでたら、無理やり中に戻し、速度を反転させる。

# Navier Stoke

要するに、流体を表す式Navier StokeをSPH用にアレンジして使っている。

まず、上のアルゴリズムの(A)は、以下の密度に関する式に基づいている。

次に、(B)は、以下の圧力に関する式に基づいている。

なお、圧力は、以下の式で計算する。

ただし、これについては、研究者によっていろいろな式が試されているようだ。例えば、以下の式を使うこともあるようだ。

ちなみに、この式で、とおけば、上と同じになる。

また、(C)は、以下の粘性に関する式に基づいている。

なお、この粘性に関する式は、粒子の速度が、周辺の粒子の速度に近づくように力が働くことを意味している。なるほどだね！

# カーネル

主に以下の３つのカーネルが使用される。

# ナブラ

またまた混乱している。SPHでは２回ほどナブラを計算する。１つ目はAlgorithmの(B)だ。つまり、を計算する必要がある。これをどうやって計算するのか？

まず、だよね。なので、

また、カーネルの式において、。両辺をで偏微分すれば、

よって、

以上より、

もう１個は、Algorithm内の(c)だ。ここでは、を計算する。はラプラシアンと呼ばれ、だよね。

まず、

次に、

結局、

# その他

質量は、以下の式で最初に定義し、シミュレーションの間は変えてはいけない。

Summationでは、自分自身のparticle も含める！