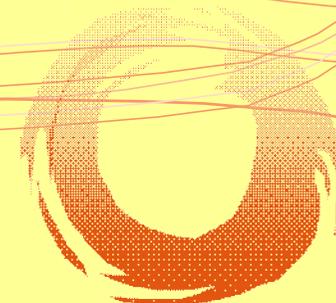


interMixingFoamを用いたタンクでの塩水混合解析(その3)

TM

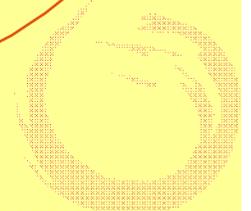


これまでの経緯

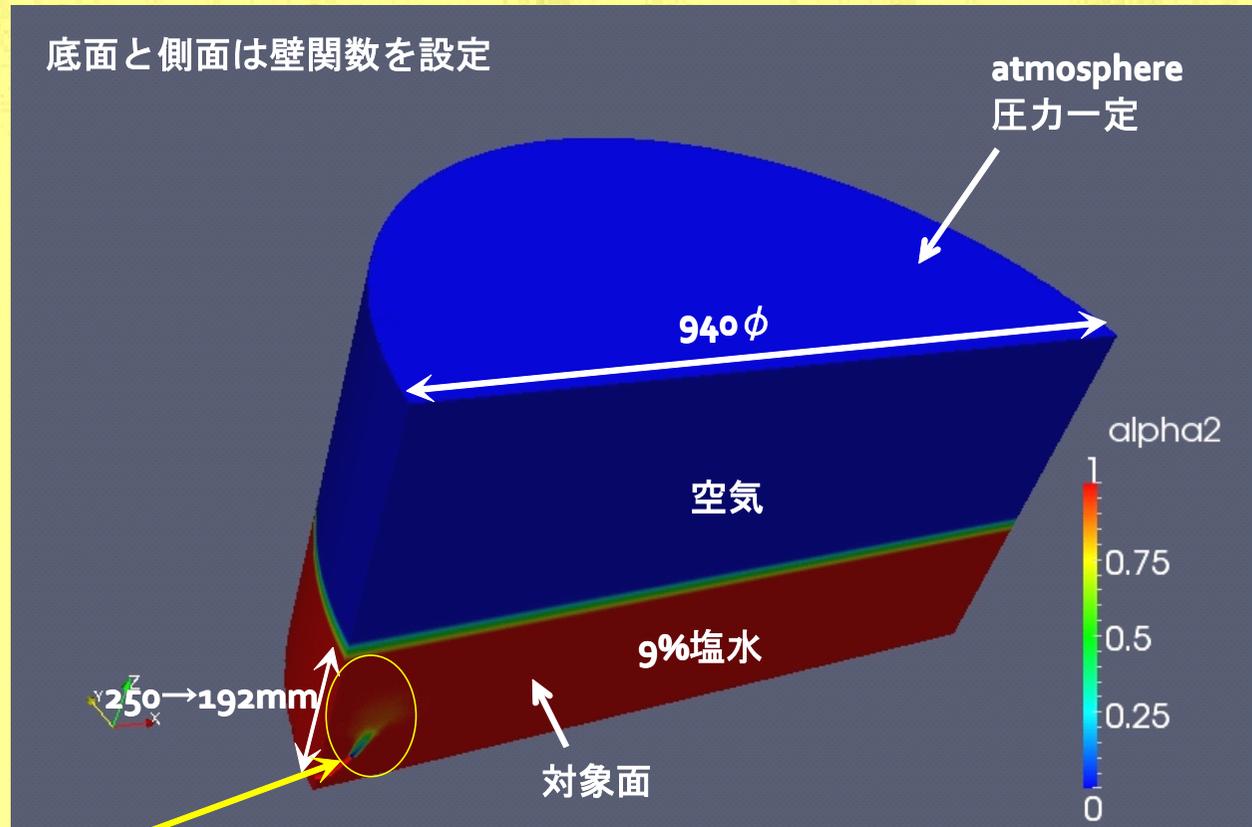
- <その1: 第11回勉強会>
計算領域すべてのstlファイルを用意。SnappyHexMeshで領域一体のメッシュ作成する方法(一体法)を紹介
→メッシュ作成方法が煩雑
- <その2: 第12回勉強会>
NE法(blockMeshで半円筒を作成し、詳細形状の境界をstlファイルで用意してsnappyHexMeshを適用)でトライアル
→未だ計算量が膨大。
- <その3: 今回>
計算環境やパラメータについて検討を行ったので報告

計算環境

- DELL studio XPS
intel Core i7 930@ 2.8GHz × 6, 12GB Memory,
1TB HDD
- Windows7 Professional SP1
- VMware Workstation 8.0.0上でDEXCS2011
- DEXCS2011上にOpenFOAM ver.2.1.0



3次元解析

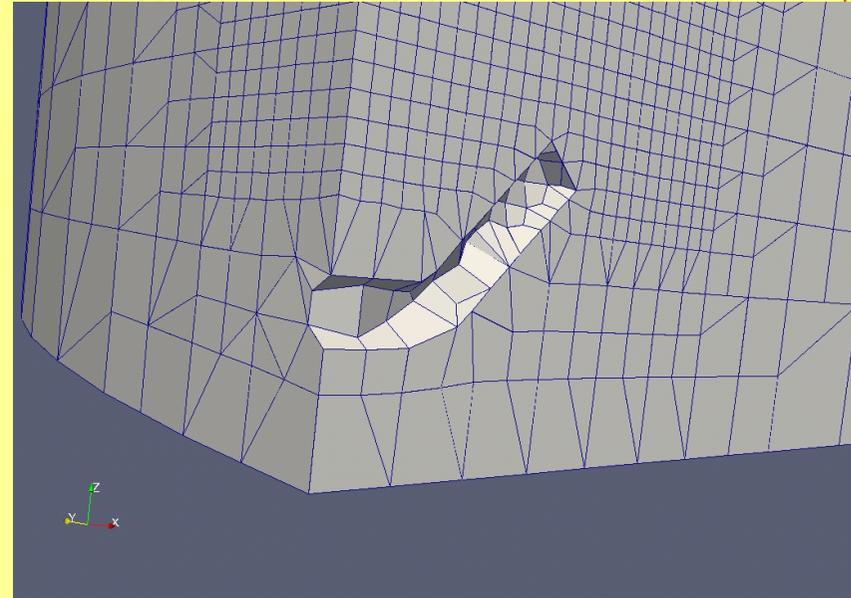
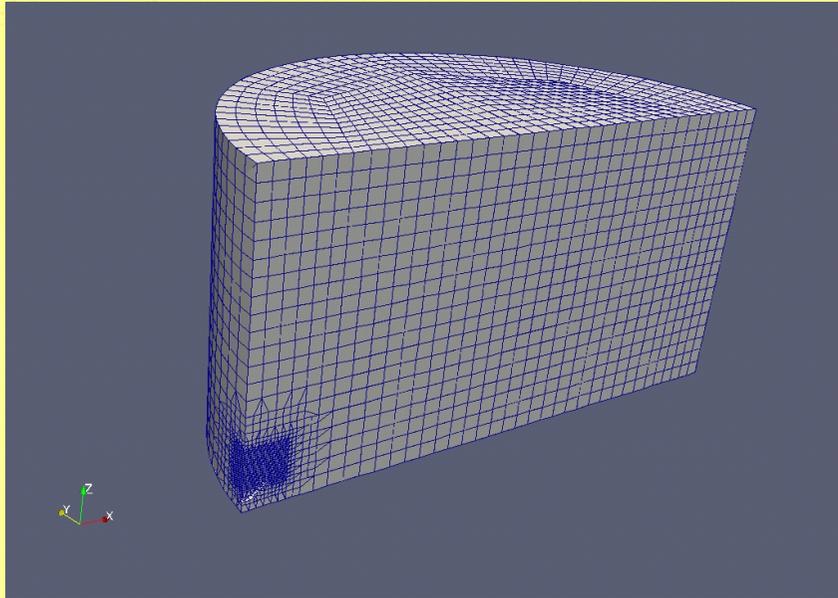


噴流なので
乱流解析が
必要

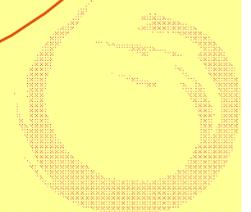
塩水タンクへの純水の注水をシミュレーション
3次元非定常層流解析
(ノズル径8φ、ノズルでの平均流速0.8m/s)



mesh



25,711cells ノズル出口形状は粗い



境界条件の変更(nut)

atmosphereで乱流粘性係数nutが過大となるので強制的に値を0.0に変更

```
atmosphere
{
  type          calculated;
  value         uniform 0.01;
}
```

← これまではkと ϵ から計算



```
atmosphere
{
  type          fixedValue;
  value         uniform 0.0;
}
```



fvsolution:収束条件の変更

- alpha tolerance 10^{-6} → 10^{-10}
- pcorr tolerance 10^{-10} → 10^{-8}
- p_rgh tolerance 10^{-7} → 10^{-9}
- p_rghFinal tolerance 10^{-7} → 10^{-11}
- (U/k/epsilon) tolerance 10^{-6} → 10^{-8}
- (U/k/epsilon)Final tolerance 10^{-8} → 10^{-10}

安定性改善。maxCo 0.2 → 0.5



decomposeParDict:領域分割方法の検討

```
numberOfSubdomains 4;  
method simple;
```

←領域分割数 **4分割**
←分割方法 **simple**を選択

- ① **simple** 簡単な領域分割
- ② **hierarchical** 基本はsimpleと同じ
だが分割する方法に順番をつける
☆他に**scotch,metis**などのCPU間
に重みをつけるものもある。

```
simpleCoeffs
```

```
{  
  n (4 1 1);  
  delta 0.001;  
}
```

← **simple**を選択した場合のパラメータ

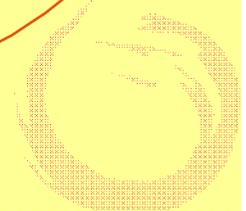
← (x y z) x方向に4分割

```
hierarchicalCoeffs
```

```
{  
  delta 0.001;  
  n (4 1 1);  
  order xyz;  
}
```

← **hierarchical**を選択した場合のパラメータ

← 分割方向の順番



領域分割条件とその結果

n	method	(x y z)	order	E-Time	C-Time
1	-	-	-	218.1	223
2	simple	(2 1 1)	-	127.1	267
2	simple	(1 2 1)	-	131.0	267
2	simple	(1 1 2)	-	126.2	260
4	simple	(4 1 1)	-	79.1	322
4	simple	(1 4 1)	-	88.2	358
4	simple	(1 1 4)	-	80.8	329
4	simple	(2 2 1)	-	101.7	415
4	simple	(1 2 2)	-	98.8	403
4	simple	(2 1 2)	-	97.0	394
4	hierarchical	(4 1 1)	xyz	78.8	320
4	hierarchical	(4 1 1)	xzy	78.8	321
4	hierarchical	(4 1 1)	yzx	78.8	320
6	hierarchical	(6 1 1)	xyz	62.8	383

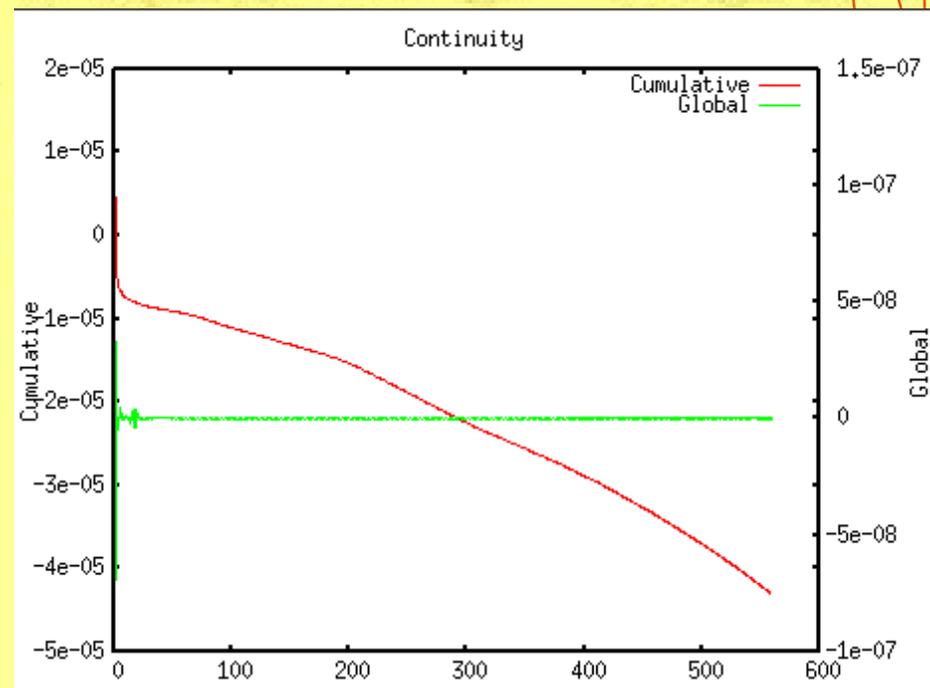
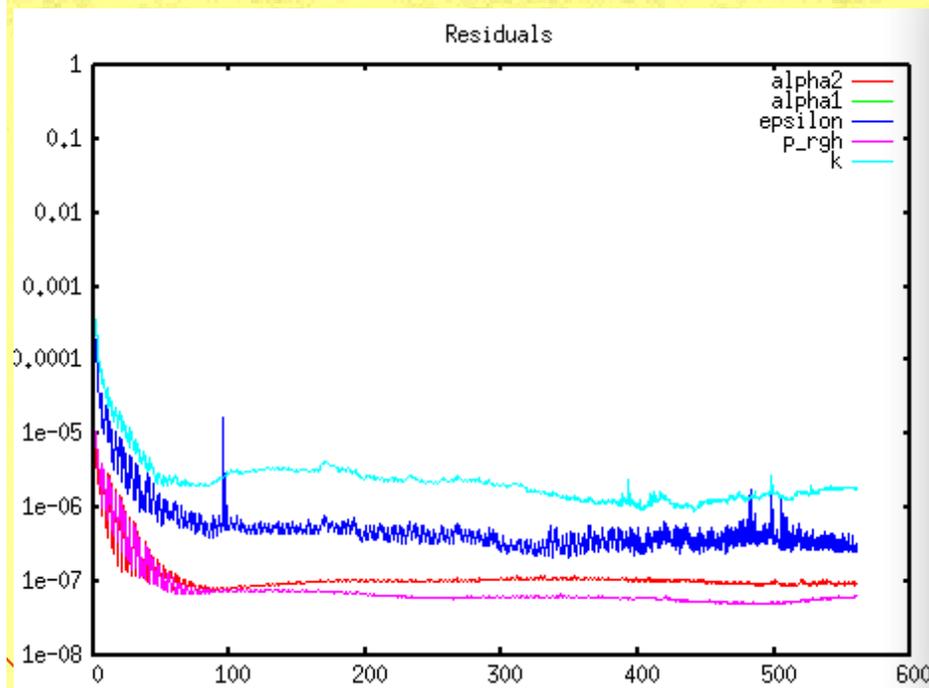
25,711cellsについてt=1sec計算。nの増加でExcutioTimeは短縮されるが、ClockTimeは増加（VMwareの影響）。xかz方向に分割をまとめた方が高速。

計算環境の変更

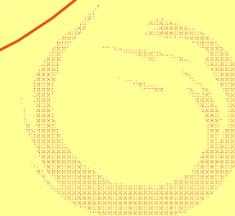
- WubiでUbuntu10.04をインストール
- Ubuntu11 (natty)に更新
- OpenFOAM ver.2.10をインストール



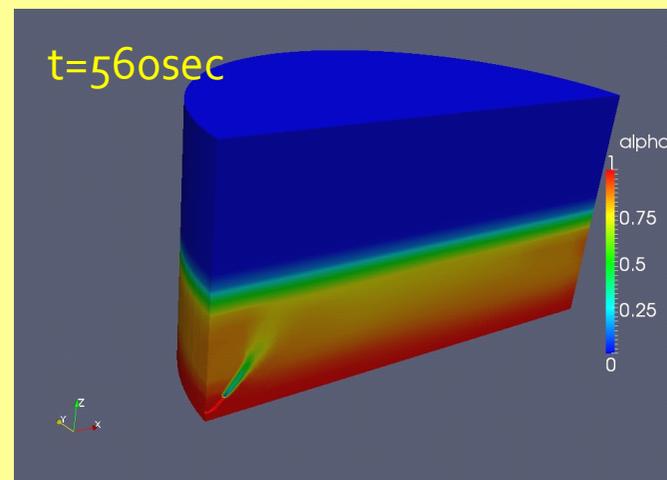
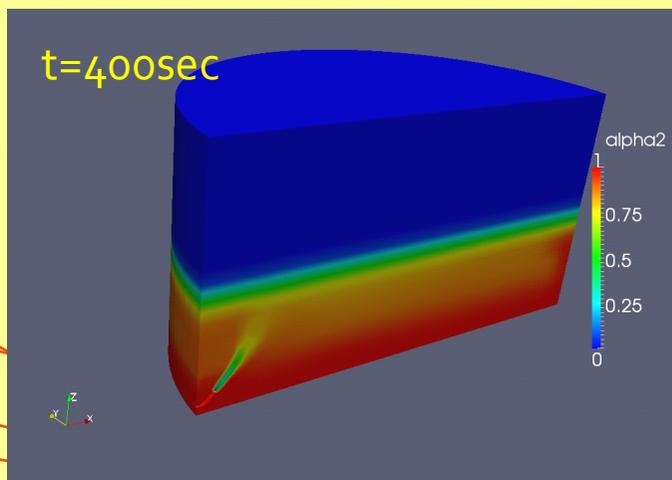
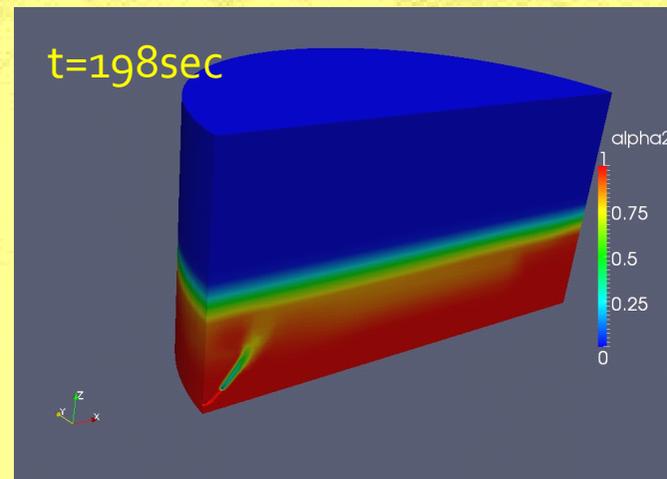
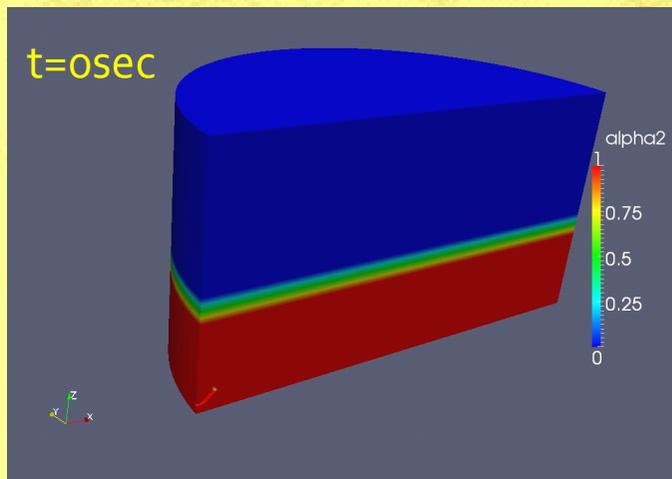
残差と保存性



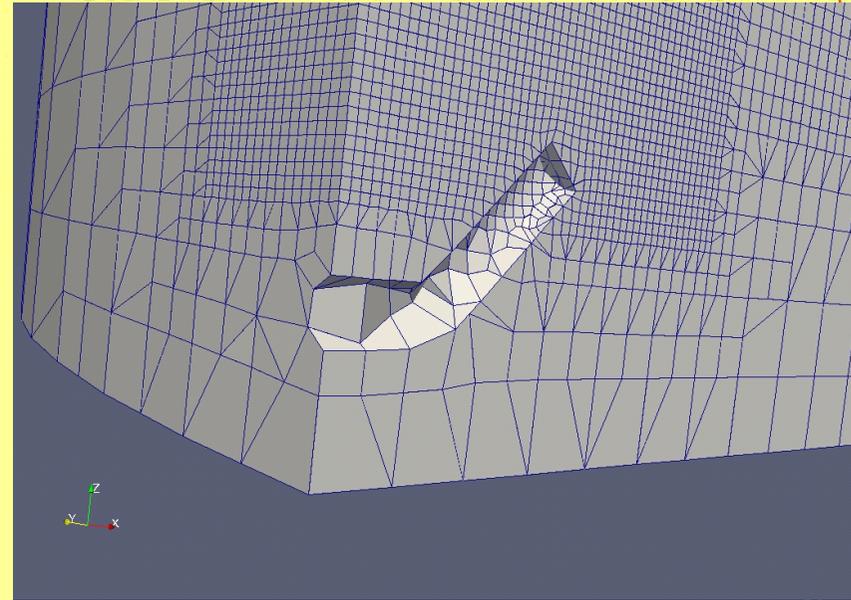
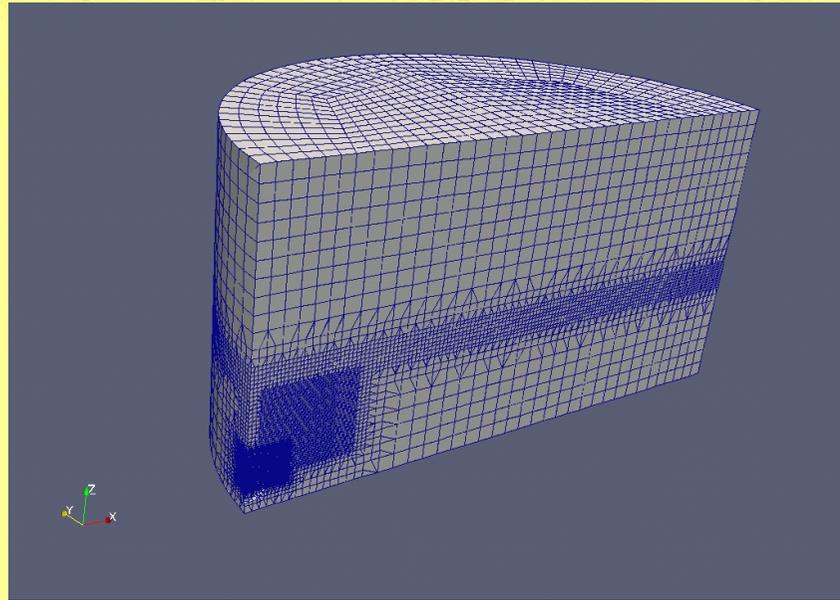
t=560secまで安定に計算終了: 14時間46分31秒
並列計算してもExecutionTimeとClockTimeはほぼ同じになった。
PyFoamPlotRunnerを使っても計算時間はそれほど増加しない。



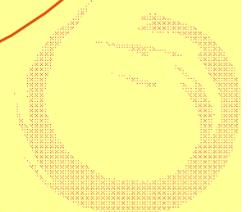
鹽水質量分率



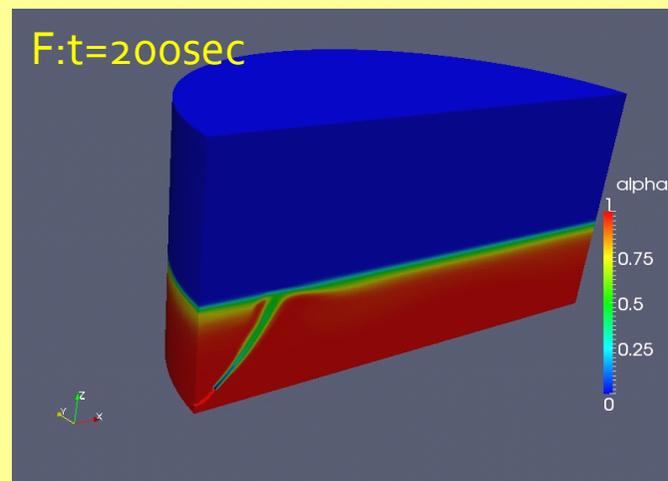
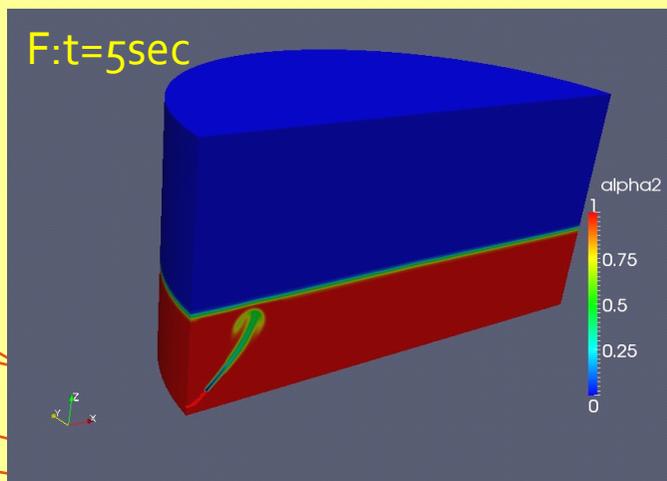
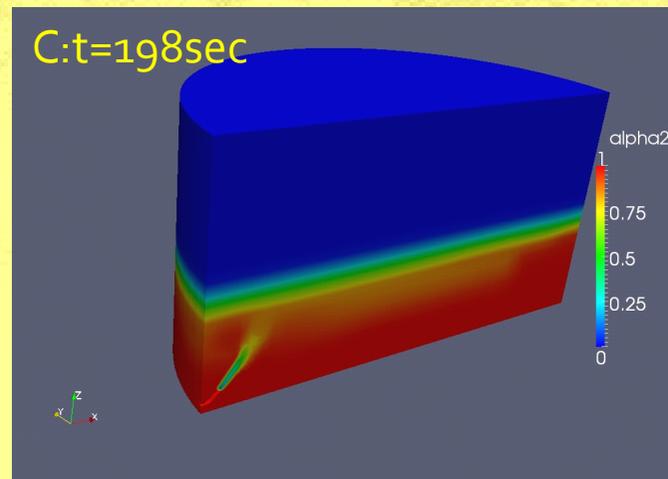
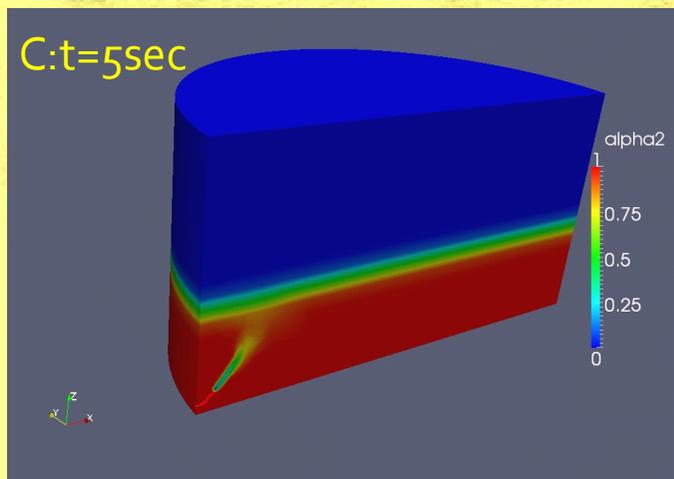
詳細メッシュ



261,698cells ノズル出口形状は改善



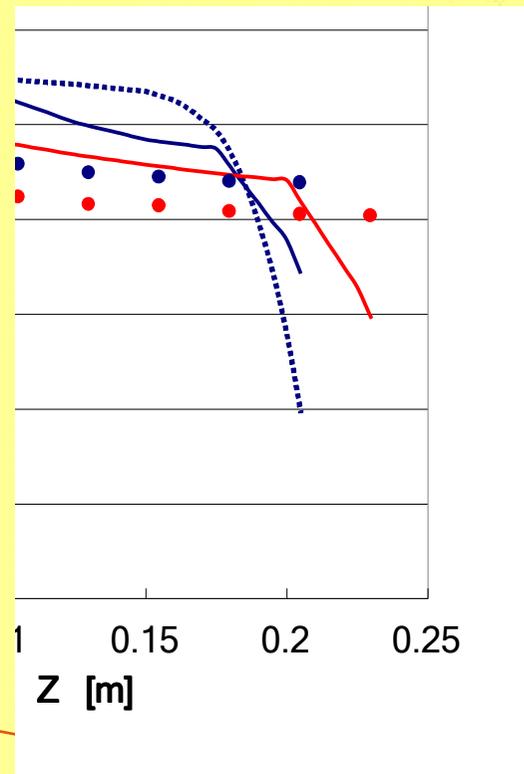
塩水質量分率



明らかに濃度分布が異なる。

実験との比較

タンク中心軸の密度分布を比較
(198secと560sec)



・計算では、液面近くに軽い液が存在しているのに対し、実験では液面近くでは混合は終わっている。

・実験値には粗いメッシュの方が近い値となっている。

・interMixingFoamには乱流濃度拡散の影響が含まれていないので、その効果を導入する必要があるものと考ええる。

おわりに

- **InterMixingFoam**を用いてタンク内の塩水混合解析について収束条件、境界条件を見直して計算安定化、精度を検討した。
- **VMware**で並列計算や**pyFoamPlotRunner**を使用すると計算量が増加。→**研究会でVMwareの設定がcpu数が1であるとの指摘あり、cpu数を増やしたら解決**
- 実験値と比較したが、粗いメッシュの方が近い値を示した。乱流濃度拡散の効果を導入する必要があるものと考えられた。