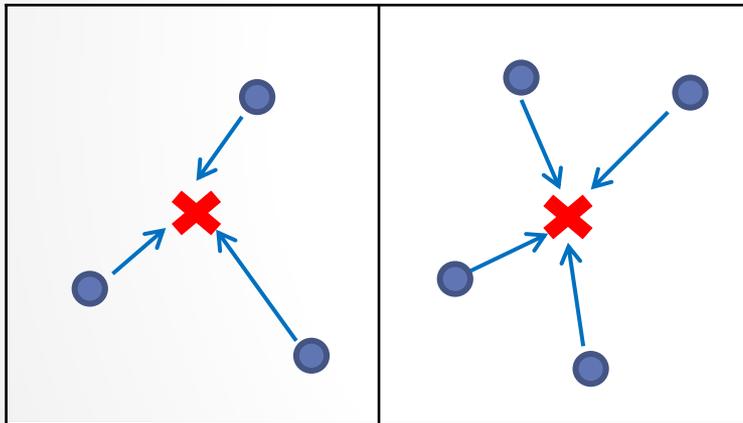


OpenFOAM-2.3.0の新機能 ⇒ Discrete Particle Modelling

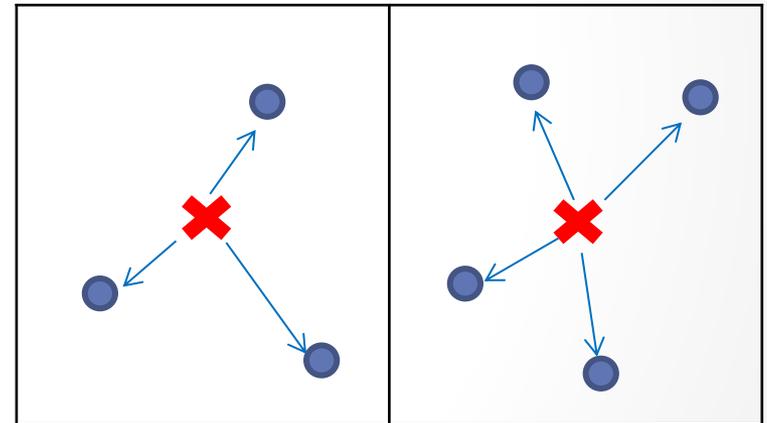
multiphase particle-in-cell method (O'Rourke 1997)

・・・連続相とラグランジュ粒子を相互に計算する古典的な解法のParticle In Cell(PIC)法より格段に安定性が良い

①各粒子から固定メッシュへ粒子特性へ補間関数使って、マッピング



②連続相の微分項を評価後、固定メッシュから各粒子へ粒子特性へ逆マッピング



*粒子割合が多い5%以上の体積分率における流れ場において、安定に計算可能かつ計算コスト低く、精度良好

*実機スケール(数十mオーダー?)の化学プロセスシミュレーションに使える

固気混相流ソルバ（流れ場は非圧縮性）

DPMFoam

MPPICFoam

- 基本的には両手法とも同じ ⇒ 異なるのは粒子の衝突の取り扱い

DPMFoam

→ 粒子衝突をバネ-ダッシュポット-スライダで表現、**DEM的扱い**
(粒子特性で剛性が必要であったり、干渉距離の閾値)

→ 体積分率大で精度良、ただし計算コスト大！(MPPICFoamの約10倍)

MPPICFoam

→ 粒子衝突を固定メッシュ上での平均値から計算、計算コスト低
(粒子特性で必要なのは質量密度だけ、簡略化)

MP-PIC法の制限

- 粒子は球形を仮定、非球形粒子では精度良く表現できない
- 固定メッシュに対して粒子が小さくならない

新ソルバのCase File その1

0

U.air、p、k.air、epsilon.air、mu.air

MPPICFoam

constant

system

transportProperties turbulenceProperties.air
g kinematicCloudProperties

controlDict、fvSchemes、fvSolution

turbulenceProperties.air

```
simulationType LES;  
                // RAS; // laminar;  
LES  
{  
  LESModel kEqn; // Smagorinsky  
  
  delta      cubeRootVol;  
  cubeRootVolCoeffs  
  {  
  }  
}  
  
RAS  
{  
  RASModel kEpsilon;  
}
```

乱れは連続相のみで
モデリング?

kinematicCloudProperties

```
solution  
{  
  active      true;  
  coupled     true;  
  transient   yes;  
  cellValueSourceCorrection on;  
  
  averagingMethod basic;  
  // moment; // dual;  
}  
  
constantProperties  
{  
  rho0      1000;  
  alphaMax  0.9;  
}
```

```
particleForces  
{  
  ErgunWenYuDrag  
  //WenYuDrag | PlessisMasliyahDrag  
  {  
    alphac alpha.air;  
  }  
}  
  
injectionModels  
{  
  totalで9種類  
}  
  
patchInteractionModel  
(local | Multi | Rebound |  
standardWall)Interaction;
```

新ソルバのCase File その2

0

U.air、p、k.air、epsilon.air、mu.air

MPPICFoam

constant

transportProperties turbulenceProperties.air
g kinematicCloudProperties

system

controlDict、fvSchemes、fvSolution

kinematicCloudProperties(前頁続き)

```
packingModel implicit; // explicit;
```

```
implicitCoeffs
```

```
{  
  alphaMin 0.0001;  
  rhoMin 1.0;  
  applyGravity false;  
  particleStressModel  
  {  
    type HarrisCrighton;  
    // Lun or exponential  
    alphaPacked 0.65;  
    pSolid 5.0;  
    beta 2.0;  
    eps 1.0e-2;  
  }  
}
```

粒子間応力の算出方法

```
dampingModel relaxation; // none;
```

```
relaxationCoeffs
```

```
{  
  timeScaleModel  
  {  
    type nonEquilibrium;  
    // equilibrium or isotropic  
    alphaPacked 0.58;  
    e 0.9;  
  }  
}
```

```
isotropyModel stochastic; // none;
```

```
stochasticCoeffs
```

```
{  
  timeScaleModel  
  {  
  }  
}
```

```
heatTransferModel none;
```

```
surfaceFilmModel none;
```

```
dispersionModel none;
```

```
radiation off;
```

```
stochasticCollisionModel none;
```

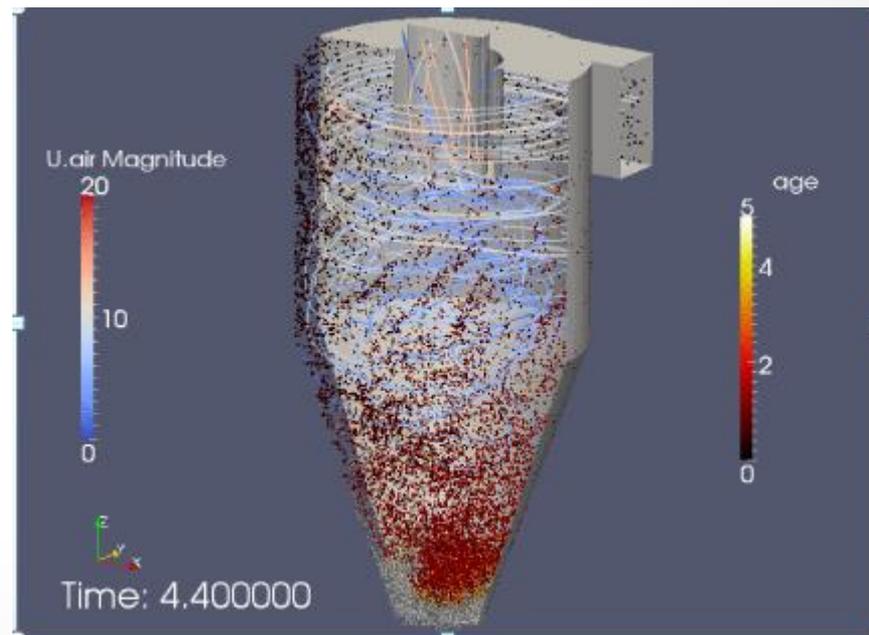
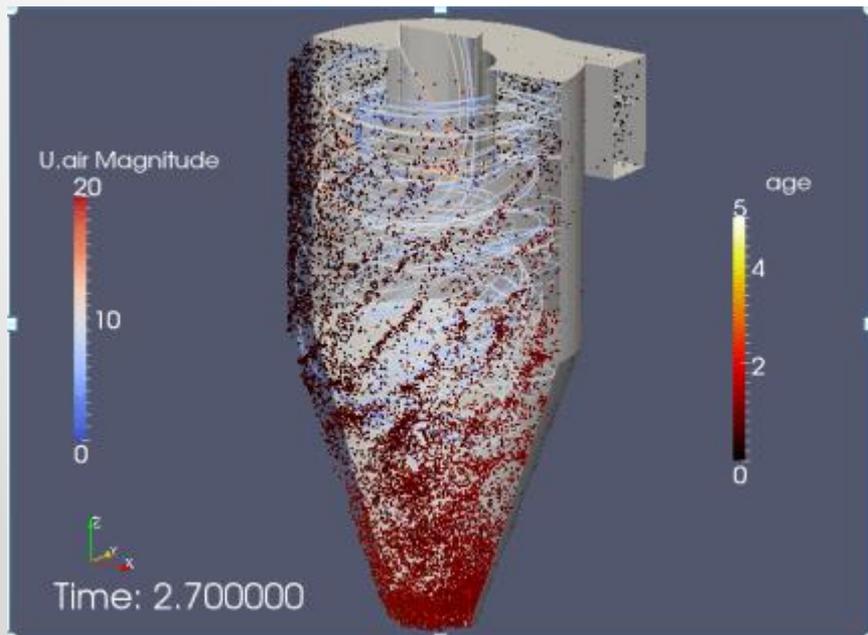
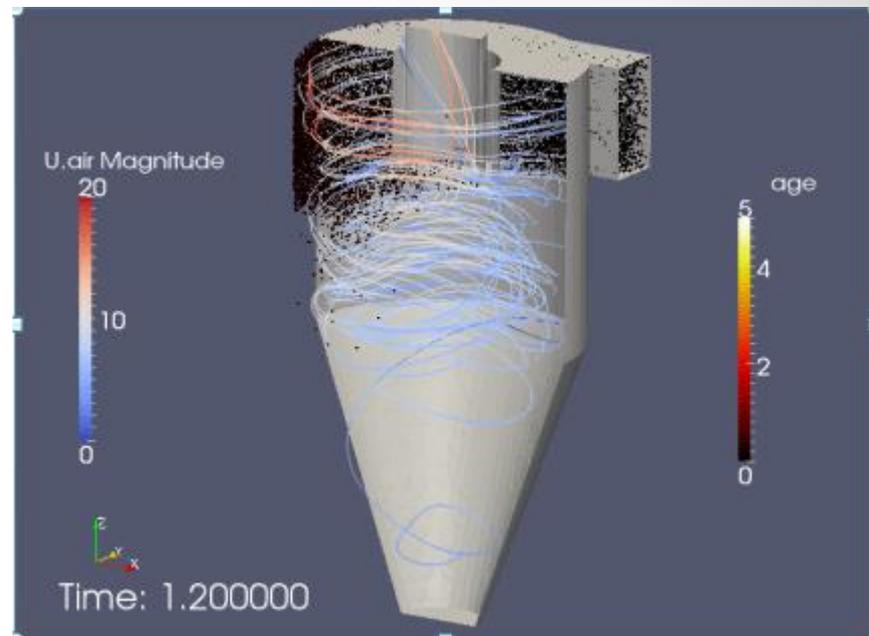
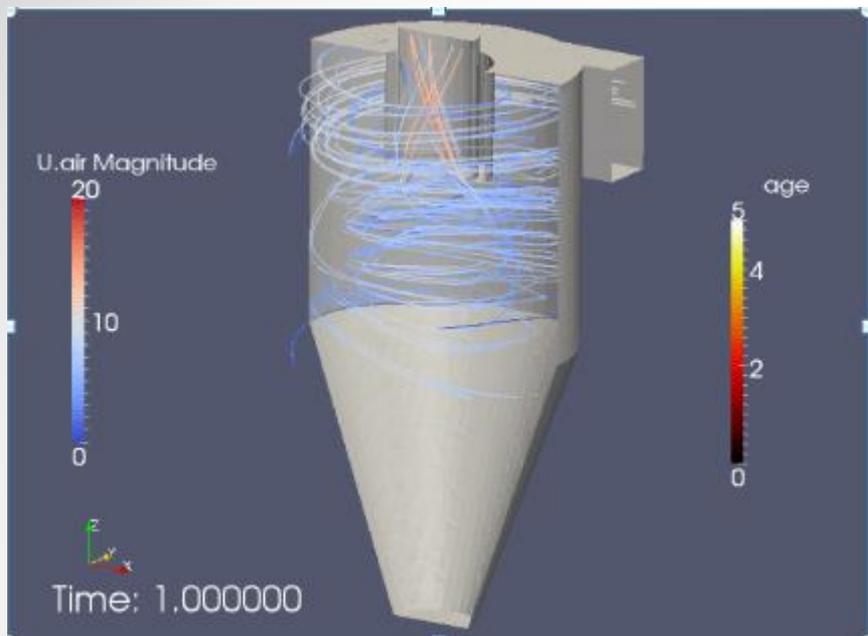
ダミー入力

```
transportProperties
```

```
continuousPhaseName air;  
rho.air 1.2;
```

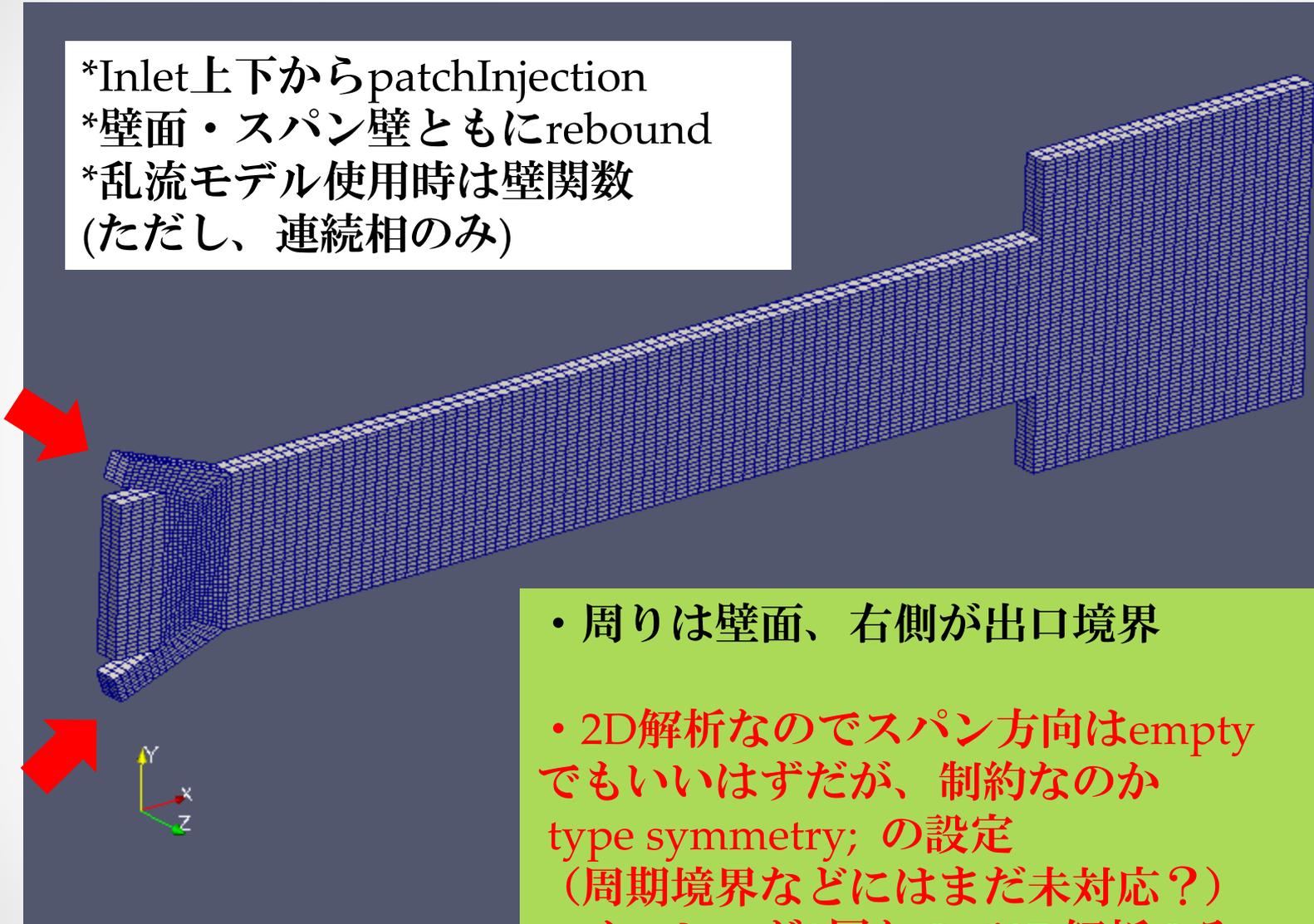
```
transportModel Newtonian;  
nu 1.568e-05;
```

解析事例 サイクロン分離器



解析事例 Cross Jet Flow

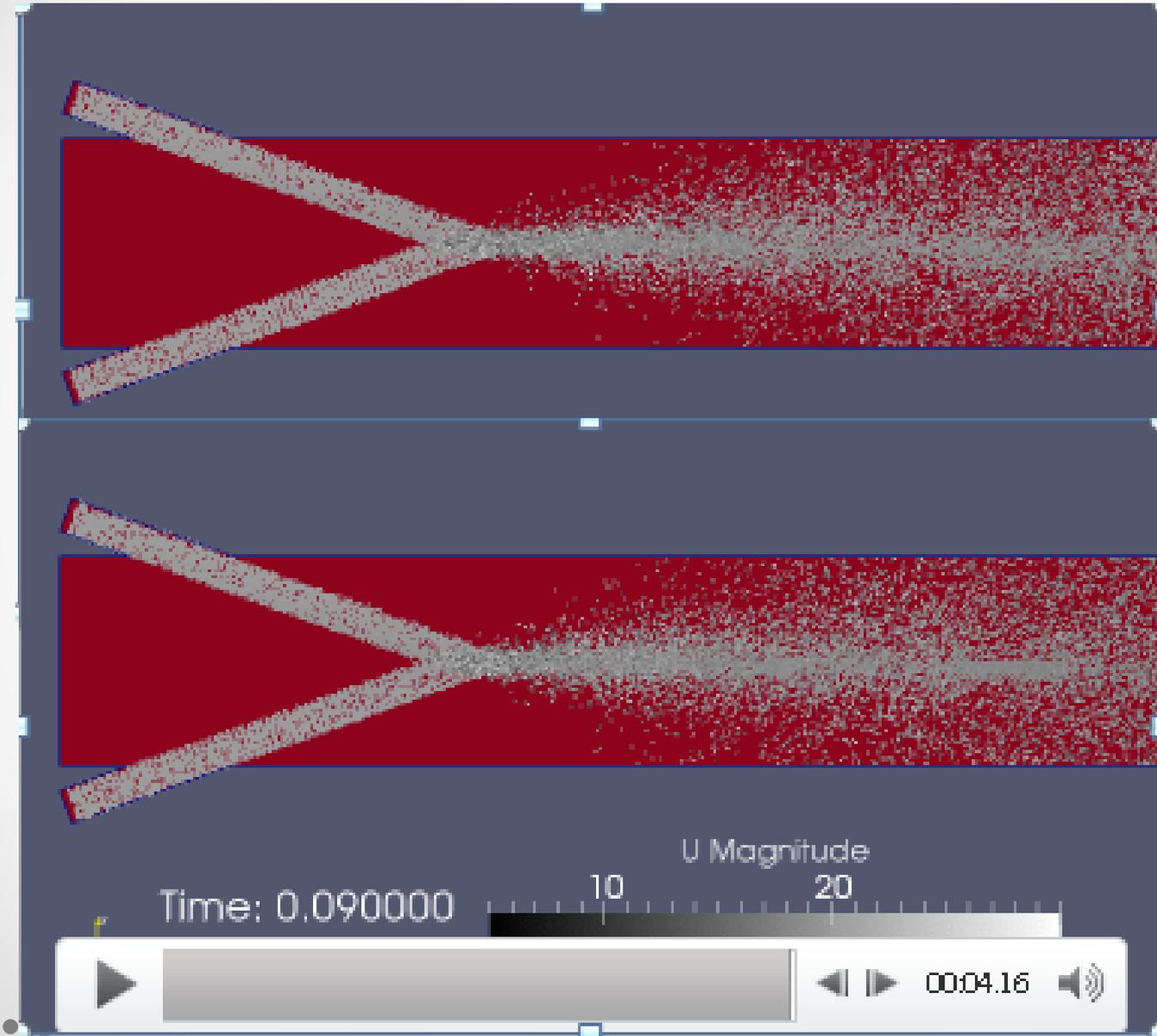
*Inlet上下からpatchInjection
*壁面・スパン壁ともにrebound
*乱流モデル使用時は壁関数
(ただし、連続相のみ)



・周りは壁面、右側が出口境界

・2D解析なのでスパン方向はempty
でもいいはずだが、制約なのか
type symmetry; の設定
(周期境界などにはまだ未対応?)
→メッシュが2層なので3D解析のみ、
2Dや軸対称は未対応の可能性有

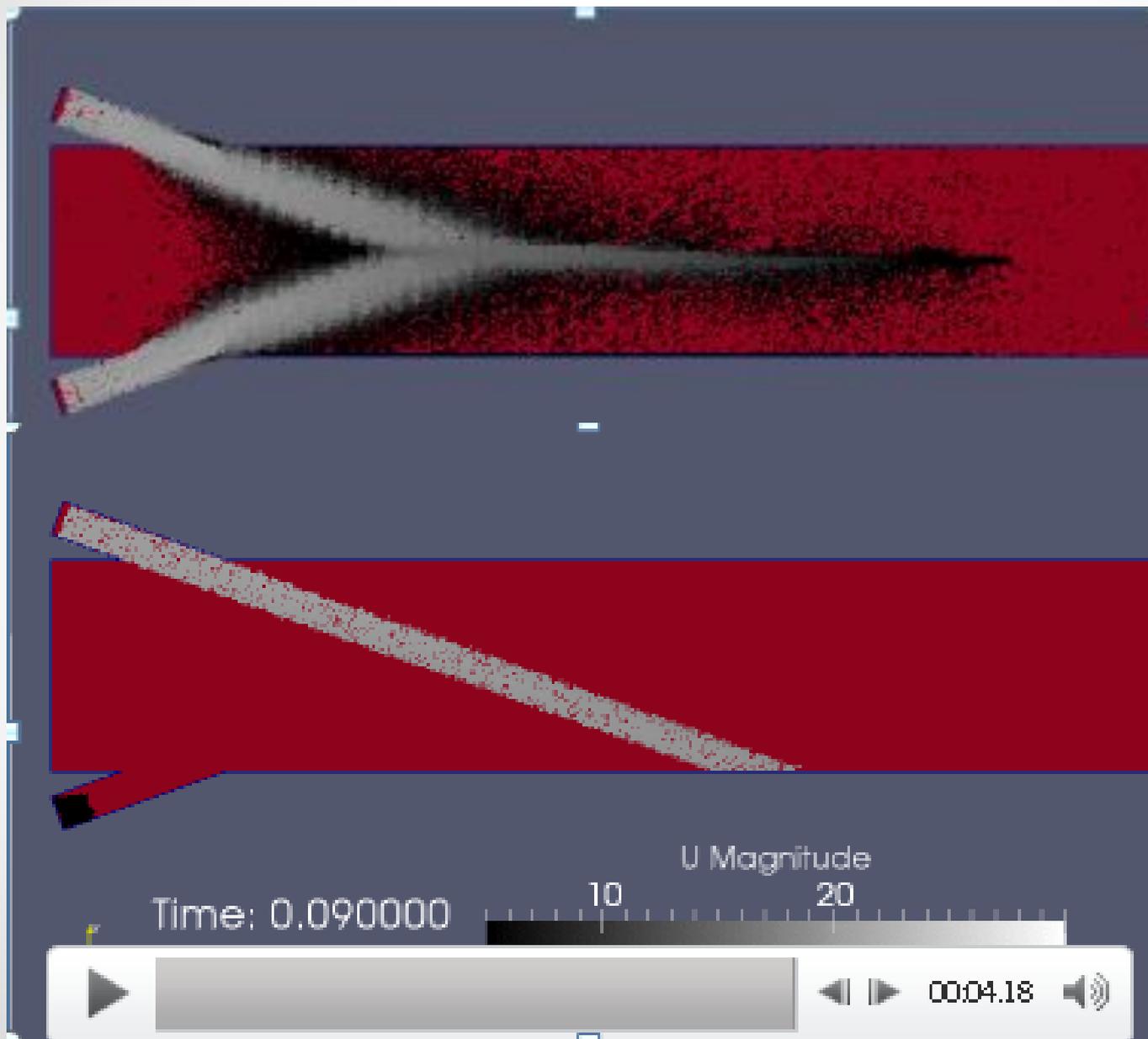
解析事例 Cross Jet Flow : 連続相の乱流モデル有無



laminar

kEpsilon

解析事例 Cross Jet Flow : 壁面挙動の差異



stick

escape