

# simpleFoamによる円管内流れ の管摩擦係数の解析

TM

第22回OpenCAE初心者勉強会(岐阜)

# 管摩擦係数

$$h = \lambda \left( \frac{l}{d} \right) \left( \frac{v^2}{2g} \right)$$

平均流速  
管摩擦係数

$$h = \frac{\Delta p}{\rho g}$$

$$\frac{\Delta p}{\rho} = \lambda \left( \frac{l}{d} \right) \left( \frac{v^2}{2} \right)$$

Darcy-Weisbachの式

ヘッド

SimpleFoamのp

# Nikuradseの実験(1933年)

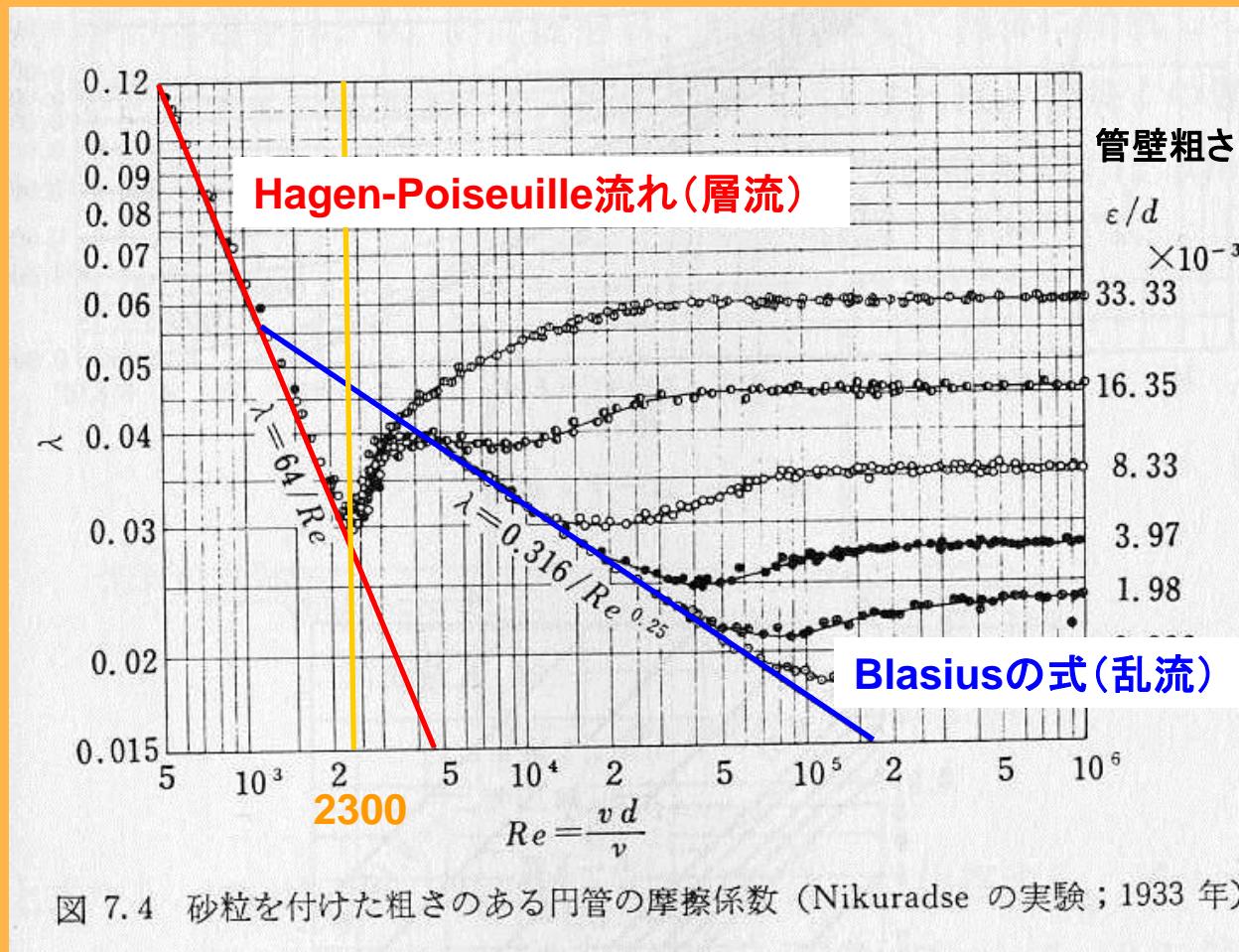
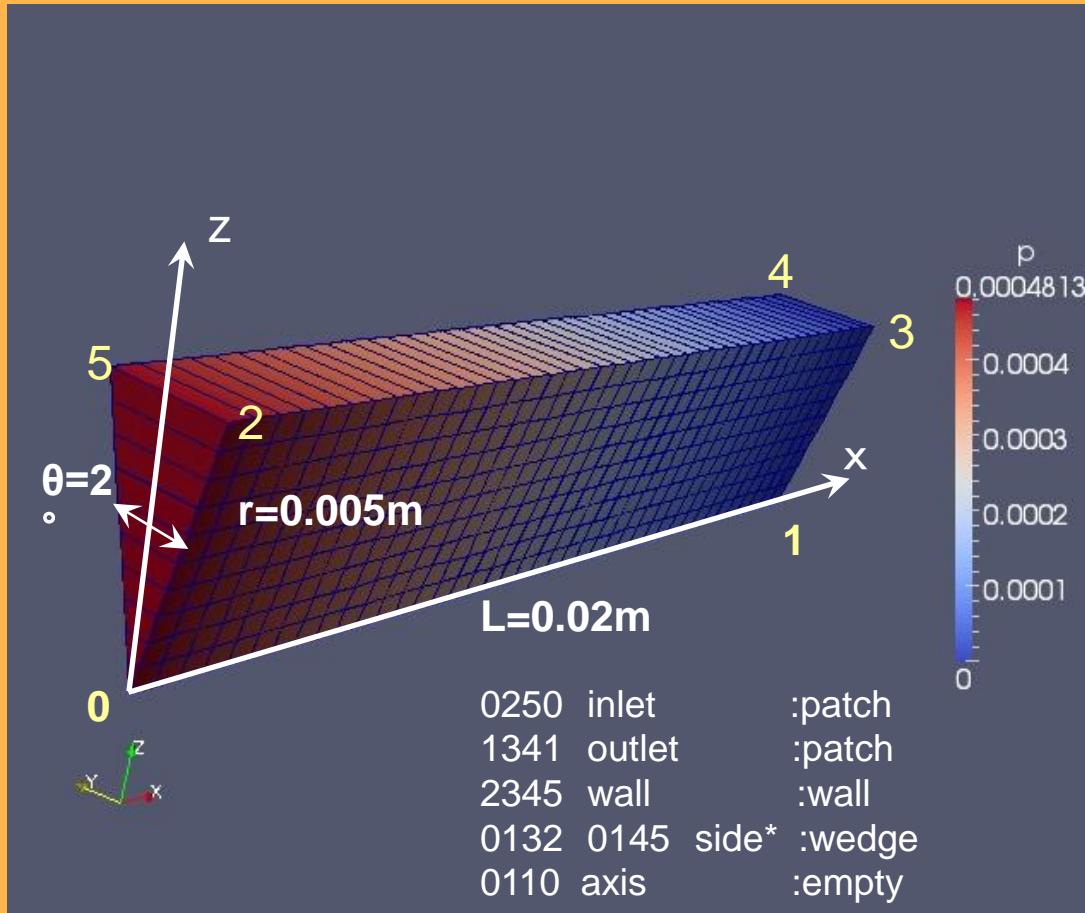


図 7.4 砂粒を付けた粗さのある円管の摩擦係数 (Nikuradse の実験；1933 年)

「流体の力学」 中山泰喜著 養賢堂 P105

# 計算モデル



# メッシュ作成

```
//- half angle of wedge in degrees [deg]
hA 30;

//- Radius of pipe [m]
r 0.005;

//- Length of pipe [m]
l 0.02;

radhA      #calc "degToRad($hA)";
yp          #calc "$r*sin($radhA)";
yn          #calc "-1.0*$yp";
z           #calc "$r*cos($radhA)";

vertices
(
    (0.0 0.0 0.0) //0
    ($l 0.0 0.0) //1
    (0.0 $yn $z) //2
    ($l $yn $z) //3
    ($l $yp $z) //4
    (0.0 $yp $z) //5
);
```

2013/5/18

**OpenFOAM ver.2.0から  
Run-time control が使える。**  
**左のようにblockMeshDictに直接プロ  
グラム記載ができる。**  
**変数を用いて、座標指定ができる。**  
**但し、通常起動だとエラー発生。**  
**以下のような処理が必要**

[tutorials/incompressible/simpleFoam/pipeCyclic](#)

```
# This case uses the #codeStream which is disabled by default. Enable for
# just this case.
MAIN_CONTROL_DICT=`foamEtcFile controlDict`
if [ -f "$MAIN_CONTROL_DICT" ]
then
    echo "Modifying ${MAIN_CONTROL_DICT} to enable allowSystemOperations"
    # Clean up on termination and on Ctrl-C
    trap 'mv ${MAIN_CONTROL_DICT}.$$ ${MAIN_CONTROL_DICT} 2>/dev/null; exit 0' \
        EXIT TERM INT
    cp ${MAIN_CONTROL_DICT} ${MAIN_CONTROL_DICT}.$$
    echo "Enabling allowSystemOperations in ${MAIN_CONTROL_DICT}."
    sed \
        -e s/"$(allowSystemOperations[ !]*\$)([0-9]\$);"/"\$1 1;"/g \
        ${MAIN_CONTROL_DICT}.$$ > ${MAIN_CONTROL_DICT}
fi
```

# 計算条件

Re	750	1500	10000	50000
層流/乱流	層流	層流	乱流	乱流
v [m/s]	$7.53 \times 10^{-2}$	0.1506	1.004	5.020
$\lambda$	$8.533 \times 10^{-2}$	$4.267 \times 10^{-2}$	$3.164 \times 10^{-2}$	$2.116 \times 10^{-2}$
$\Delta p$ [Pa]	0.483	0.966	31.84	532.3
$\Delta p/\rho$ [ $m^2/s^2$ ]	$4.838 \times 10^{-4}$	$9.677 \times 10^{-4}$	$3.189 \times 10^{-2}$	0.5332

- ❖ 流体: 水(20°C)  $\rho=998.2(\text{kg}/\text{m}^3)$ 、 $v=1.004(\text{m}^2/\text{s})$
- ❖ Reの4つの条件について解析
- ❖ この解析では $\Delta p/\rho$ を与えて平均流速vを求める
- ❖ Hagen-Poiseuilleの理論式とBlasius式の値と解析値を比較

# simpleFoam

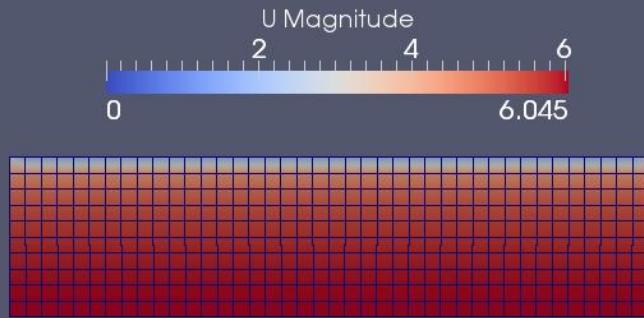
OpenFOAM ver.2.1.x tutorials

incompressible/simpleFoam

<前提>

- ・定常問題
- ・非圧縮性
- ・非圧縮性乱流モデル

# 計算条件



Mesh=40 × 1 × 10  
itartion:20000



<計算手法と収束条件>  
pipecycle default

<流体モデル>

層流: laminar

乱流: kEpsilon

<境界条件>

inlet,outlet

P: fixedValue \*

U,k,epsilon:zeroGradient

nut:calculated 0

wall

P:zeroGradient

U:fixedValue 0,0,0

k:kqRWallFunction 0

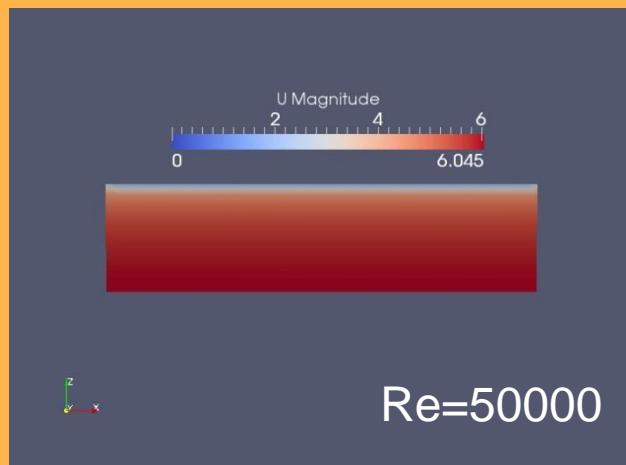
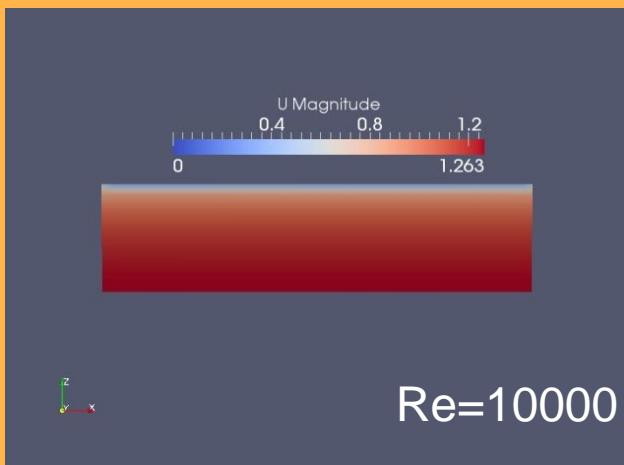
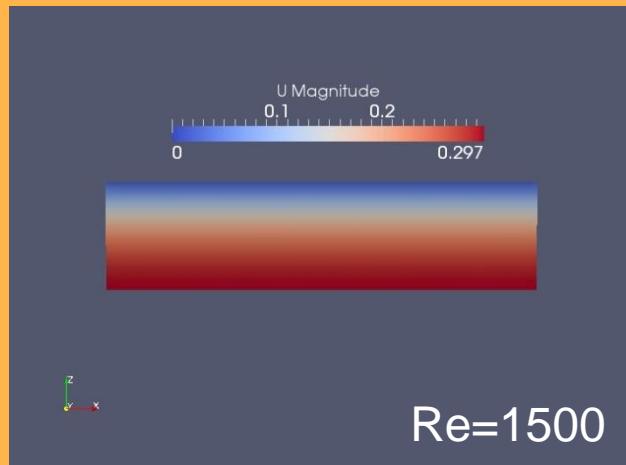
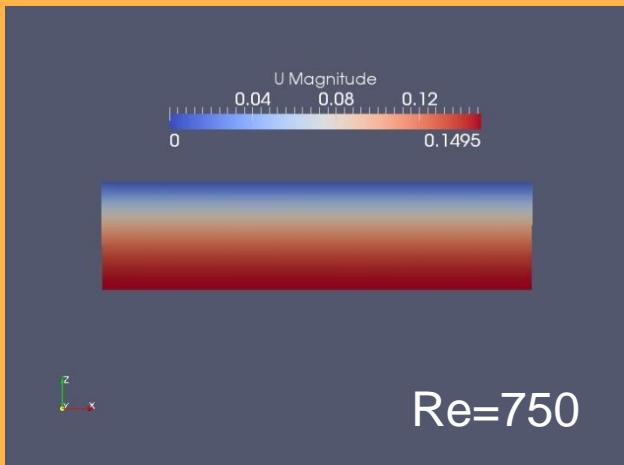
epsilon: epsilonWallFunction 0

nut:nutkwallfunction 0

side\* P,U,k,epsilon,nut :wedge

axis P,U,k,epsilon,nut :empty

# 解析結果



# 解析結果(Reの影響)

Re	750	1500	10000	50000
層流/乱流	層流	層流	乱流	乱流
v [m/s]	0.07564	0.1505	1.046	5.163
vmax [m/s]	0.1495	0.2970	1.263	6.045
λ(計算)	$8.457 \times 10^{-2}$	$4.272 \times 10^{-2}$	$2.915 \times 10^{-2}$	$2.000 \times 10^{-2}$
λ(理論)	$8.533 \times 10^{-2}$	$4.267 \times 10^{-2}$	$3.164 \times 10^{-2}$	$2.116 \times 10^{-2}$

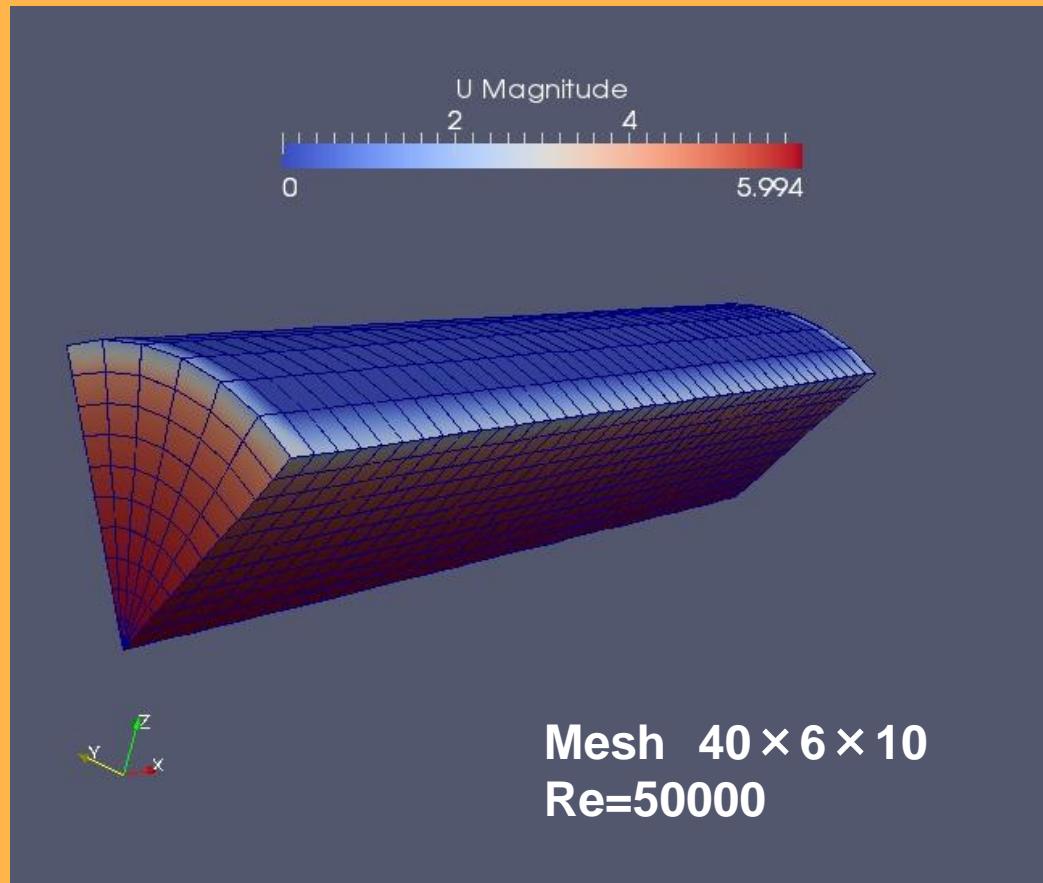
- ・平均流速vは `patchAverage -latestTime U outlet` で計算
- ・層流vmaxは理論的には平均流速の2倍。計算結果は良好
- ・乱流vmaxは実験で1.14～1.25倍の範囲。計算結果は良好
- ・計算λは約5%の精度、層流の方が精度良好

# 解析結果(メッシュ影響)

	Re=1500 (層流)		Re=10000 (乱流)		Re=50000 (乱流)	
Mesh	v [m/s]	vmax[m/s]	v [m/s]	y+	v [m/s]	y+
$40 \times 1 \times 5$	0.1526	0.2918	1.048	29.8	5.183	122
$40 \times 1 \times 10$	0.1505	0.2970	1.046	15.3	5.163	62.7
$40 \times 1 \times 20$	0.1500	0.2987	0.8661	7.7	5.167	31.8
$40 \times 1 \times 40$	0.1497	0.2987	0.7242	3.8	5.173	15.9

- ・メッシュは $40 \times 1 \times 5$ で十分
- ・y+は [yPlusRAS](#)で計算
- ・乱流の場合にy+が15以下になると平均流速vの精度が悪化
- ・対数則の壁面境界条件の場合はy+=30~100を遵守する必要あり

# 3Dメッシュ影響



平均流速  $v=5.134\text{m/s}$   
精度は2次元軸対象とあまり変わらない

# 管壁粗さの影響

		Case1	Case2	Case3	Case4
	管壁粗さ $\epsilon/d$ $[10^{-3}]$	3.970	8.330	16.35	33.33
	粗さ高 $K_s = \epsilon [m \times 10^{-5}]$	3.970	8.33	16.35	33.33
	$\Delta p/\rho$ $[m^2/s^2]$	0.63	0.8316	1.134	1.512
	$v[m/s]$	5.052	4.828	4.739	4.667
	Re	50319	48088	47201	46484
	$y^+$	68.0	78.4	91.5	105.7
	$\lambda$ [計算]	0.025	0.036	0.051	0.069
	$\lambda$ [Nikurzde]	0.025	0.033	0.045	0.060

nut境界条件で以下のように設定。結果は壁が粗くなると精度悪化

```
type nutkRoughWallFunction;
```

```
Ks uniform * ;
```

```
Cs uniform 0.5; ←均一度 0.2-1.0の範囲だが均一の場合は0.5
```