

# ただで始める流体解析

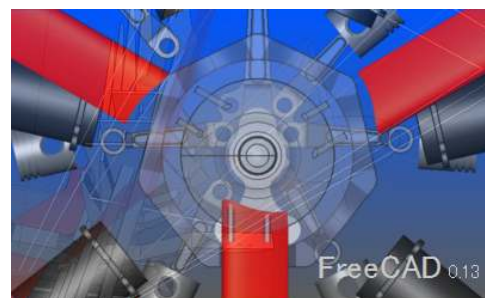
## 初心者が行う OpenFOAM & Free CAD & Blender を利用したファンの流れ計算 (MRF, AMI) Ver 2.2.x

2013.6.15

1/83

### 本日の流れ

0. はじめに
1. ファンの作成
2. チャンバー等の作成
- 3-1. メッシュの作成 (MRF)
  - 2. 計算の設定
  - 3. 計算の結果
  - 4. 特性計算の結果
- 4-1. メッシュの作成 (AMI)
  - 2. 計算の設定
  - 3. 計算の結果
5. まとめ, その他



Open  FOAM

2013.6.15

2/83

# 0. はじめに

2013.6.15

3/83

## 0. はじめに

ファンの流れ解析をOpenFOAM, FreeCAD, blenderの**すべて** **ただで使えるフリーソフト**を利用してやってみました。

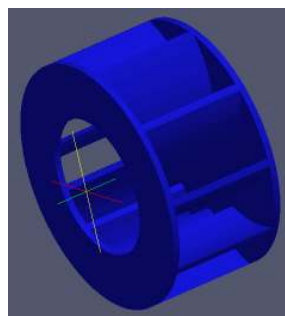
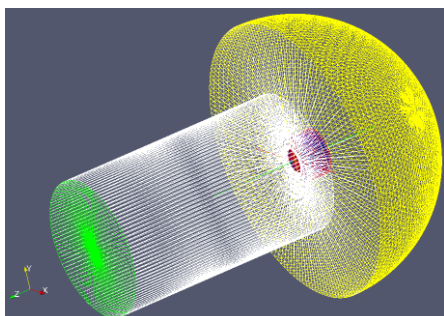


+



+

OpenFOAM



今回の計算モデル  
Free CADで形状  
を全部作成

2013.6.15

4/83

MRFの場合、回転体計算の設定が、ver2.1とver2.2では多少異なっています。

ver2.1まではMRFSimpleFoamのソルバーがありましたが、ver2.2からはsimpleFoamに統合されました(MRFSimpleFoam自体がなくなった)。使い方はチュートリアルを参照。

●ソルバー

ver2.1 MRFSimpleFoam



ver2.2 simpleFoam

●回転領域の設定

ver2.1 constant/MRFZones



ver2.2 system/fvOptions

2013.6.15

5/83

AMIを利用した計算では、回転領域作成に関してネイティブのOpenFoamの機能のみを使った場合、色々と設定のノウハウがあるようです。

①OpenFOAMのGoogle groupに色々と情報あり。

pimpleDyMFoamを使ったタービン翼内の計算に関して

②ながれ31(2012)255-262 [特集]オープンソースの大きな流れ 自動車部品開発における大規模 流体計算への取り組み

③2011/12/2 オープンCAEシンポジウム2011 オープンCAEを活用した大規模高速演算及び大規模モデルの取扱

今回は、簡単なプロペラのチュートリアルにある方法でやってみました。上記に挙げた方法は現在トライ中、またいつか報告予定。

2013.6.15

6/83

AMIの最新機能を利用して計算をするため、2.2.x版が必要！  
2.2.0のリリース後にAMIには修正が色々あります。gitで持ってきてコンパイル！

[OpenFOAM-2.2.x / src / meshTools / AMIInterpolation / AMIInterpolation /](#) [+](#)

[History](#)

Correcting maximum feature angle target walk		
Sergio Ferraris authored 2 hours ago <span style="float: right;">latest commit 0a0b63586c <a href="#">+</a></span>		
..		
<a href="#">AMIInterpolation.C</a>	2 hours ago	Correcting maximum feature angle target walk [Sergio Ferraris]
<a href="#">AMIInterpolation.H</a>	2 months ago	ENH: Merging some updates from the internal development line [andy]
<a href="#">AMIInterpolationI.H</a>	2 months ago	Initial commit for version 2.2.x [OpenFOAM-admin]
<a href="#">AMIInterpolationName.C</a>	2 months ago	Initial commit for version 2.2.x [OpenFOAM-admin]
<a href="#">AMIInterpolationParallelOps.C</a>	2 months ago	Initial commit for version 2.2.x [OpenFOAM-admin]
<a href="#">AMIPatchToPatchInterpolation.H</a>	2 months ago	Initial commit for version 2.2.x [OpenFOAM-admin]

2013.6.15

7/83

# 1. モデルの作成 ファン

2013.6.15

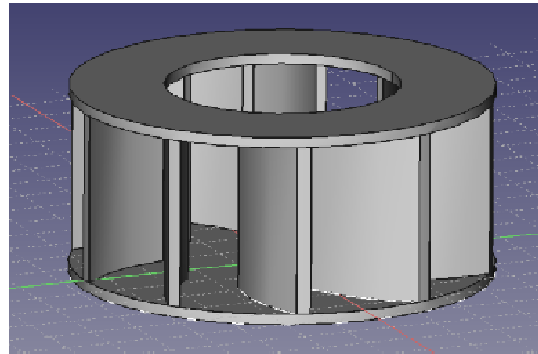
8/83



# 1. モデルの作成 ファン 概要

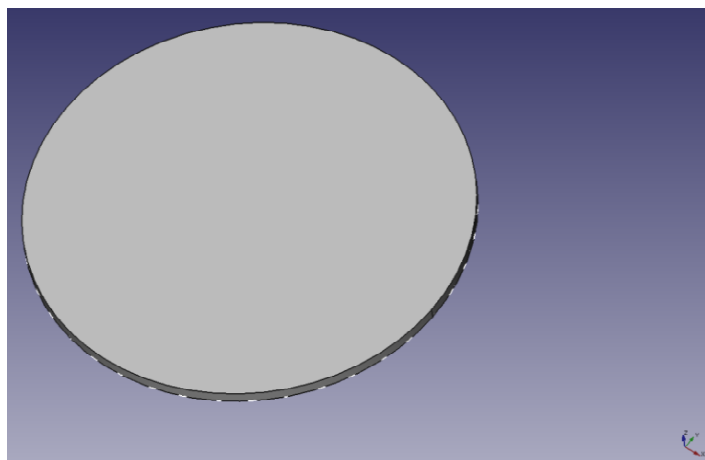
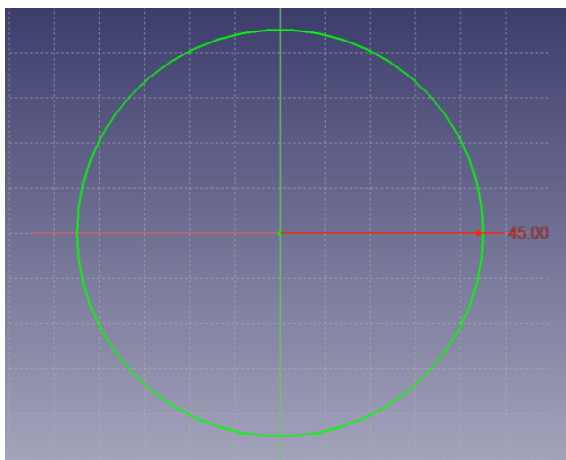


パラメータ	羽根車外径 $D_2$ mm	羽根車内径 $D_1$ mm	翼数	入口角 $\beta_1$ deg	出口角 $\beta_2$ deg
設定値	90	50	10	60	45

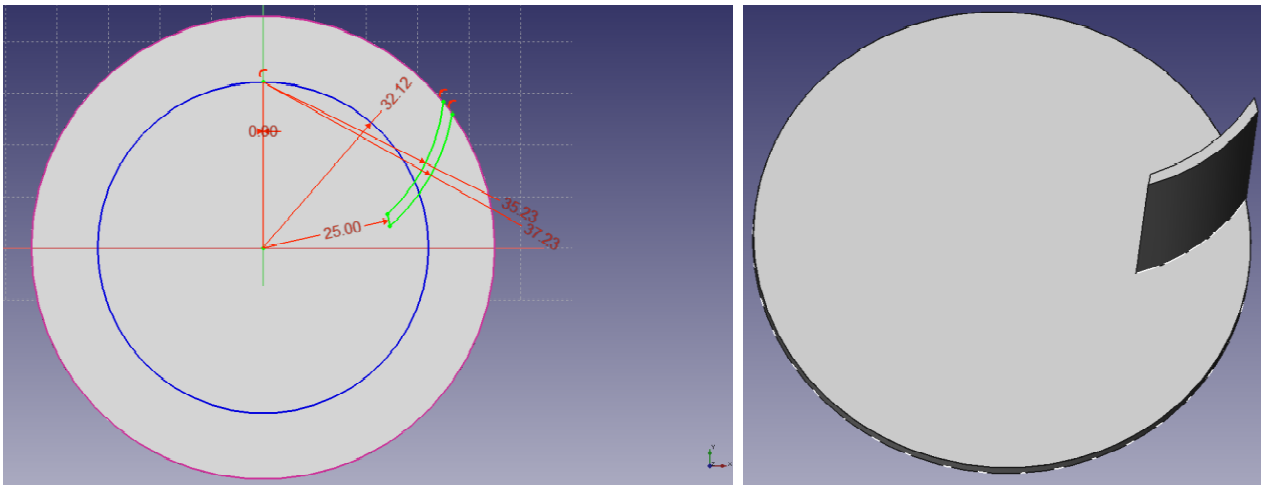
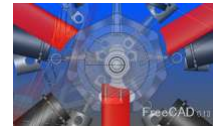


ファン高さ45

# 1. モデルの作成 ファン 底部分の作図



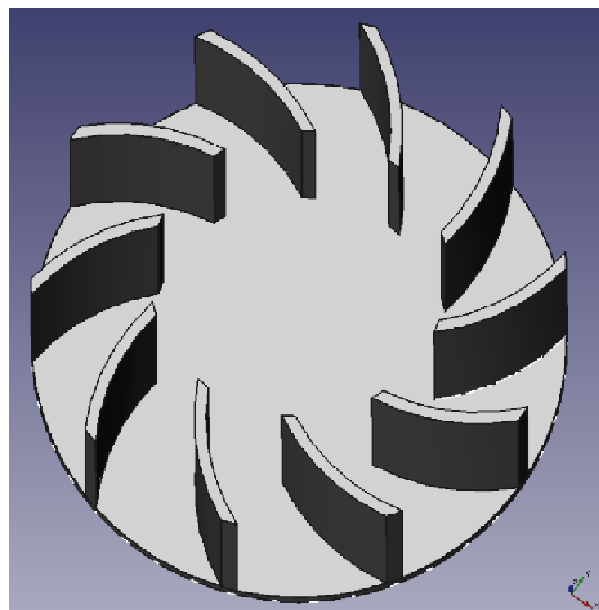
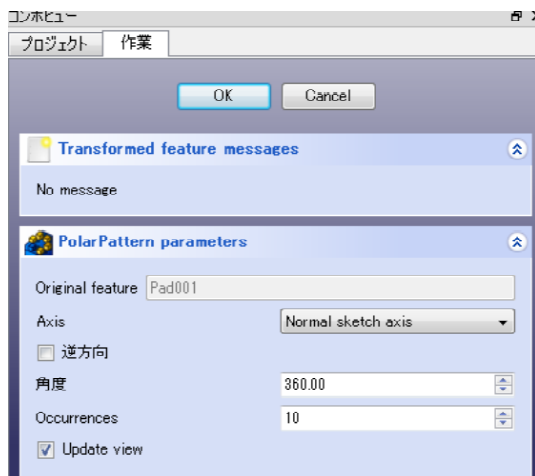
# 1. モデルの作成 ファン 羽根1枚の作図



2013.6.15

11/83

# 1. モデルの作成 ファン 羽根の回転コピー

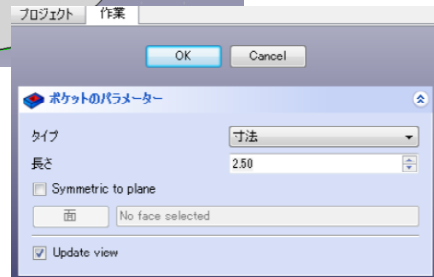
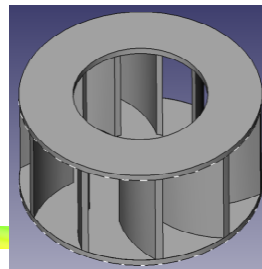
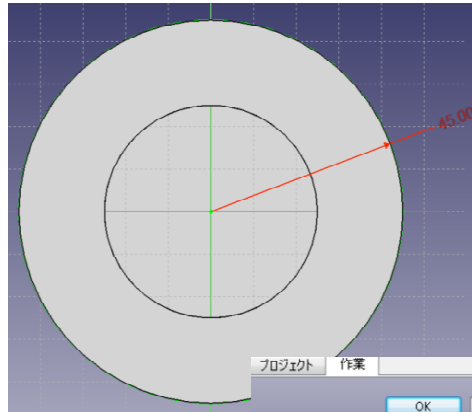
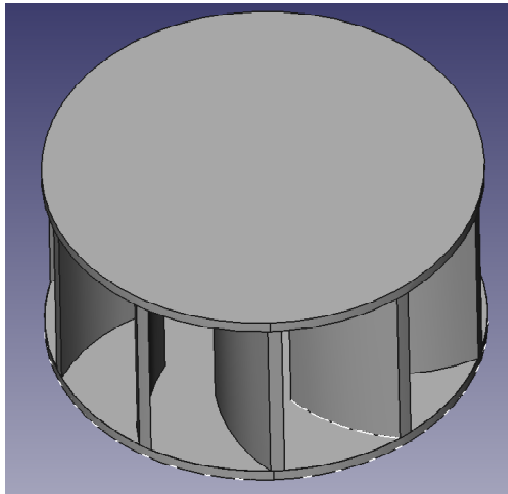


2013.6.15

12/83

# 1. モデルの作成 ファン 上部分の作図

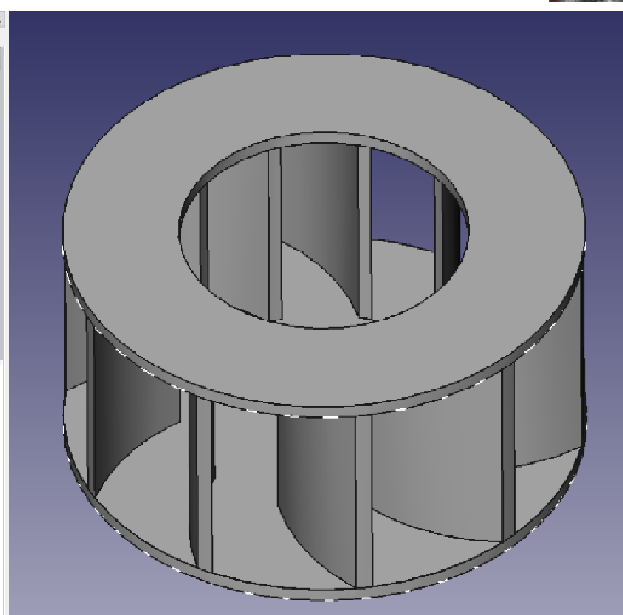
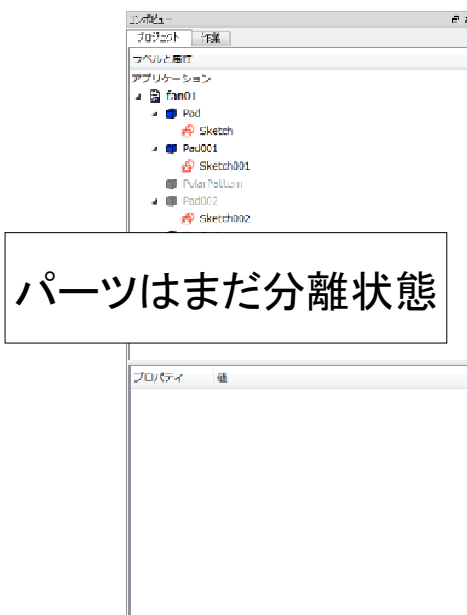
# 吸い込み口の作成



2013.6.15

13/83

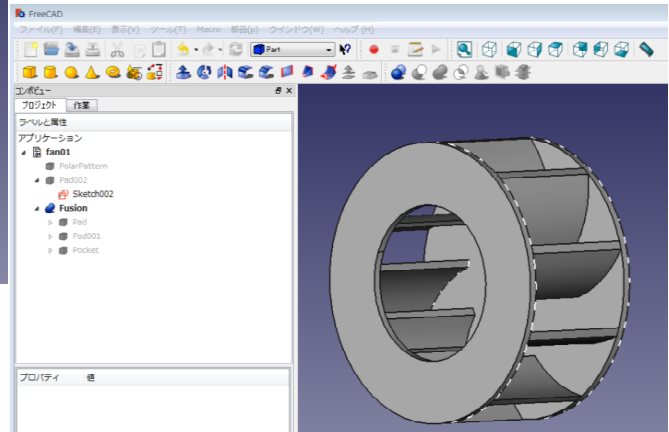
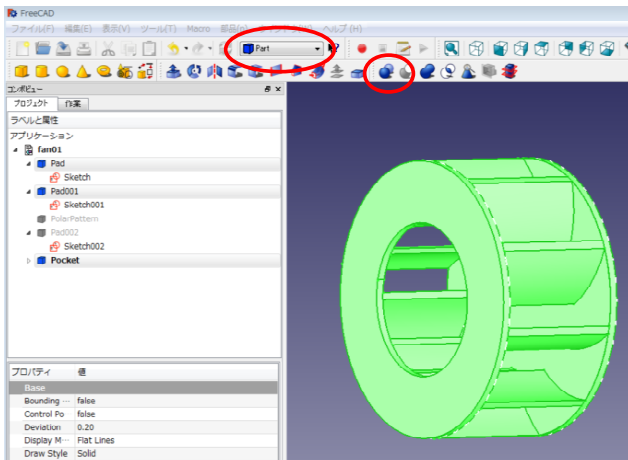
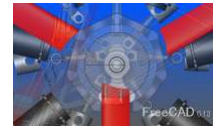
# 1. モデルの作成 ファン



2013.6.15

14/83

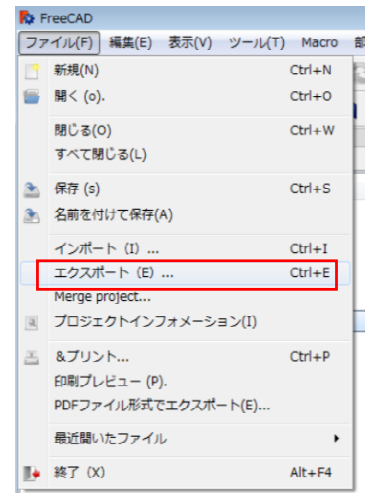
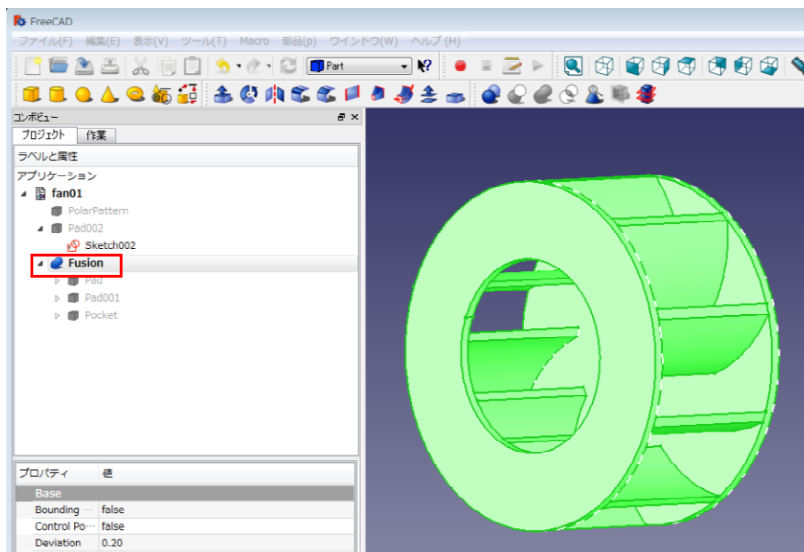
# 1. モデルの作成 ファン ブーリアン演算



2013.6.15

15/83

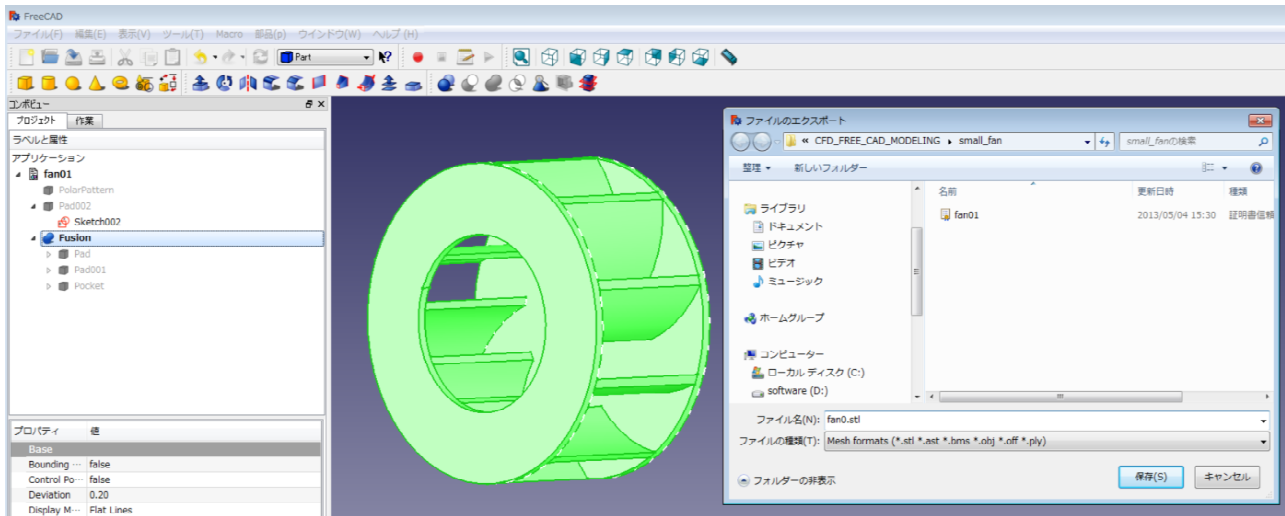
# 1. モデルの作成 ファン モデルのエクスポート



2013.6.15

16/83

# 1. モデルの作成 ファン モデルのエクスポート

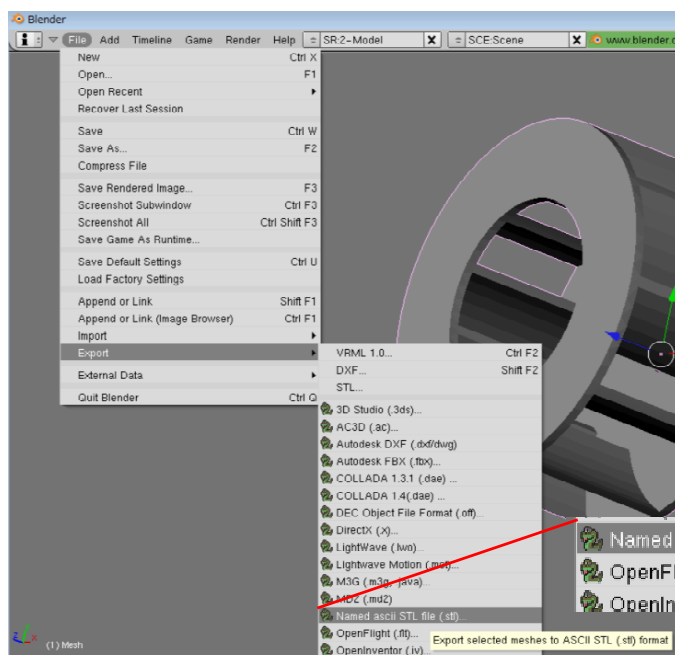


stl形式, ただしバイナリー

2013.6.15

17/83

# 1. モデルの作成 ファン blenderでnamed ascii STL形式に変換



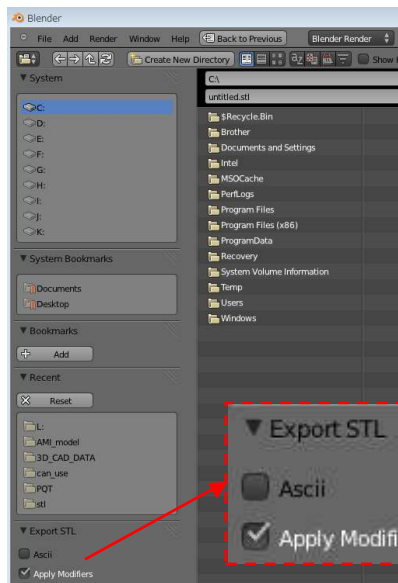
blender2.49bを利用  
使い慣れているから

2013.6.15

18/83

## 1. モデルの作成 ファン

blender2.49bでnamed ascii STL形式を書き出すためには、stlasciiexport.pyを持ってきてセットする必要あり。



新しいblenderではAsciiでSTL形式を書き出す設定が初めからあり。



2013.6.15

19/83

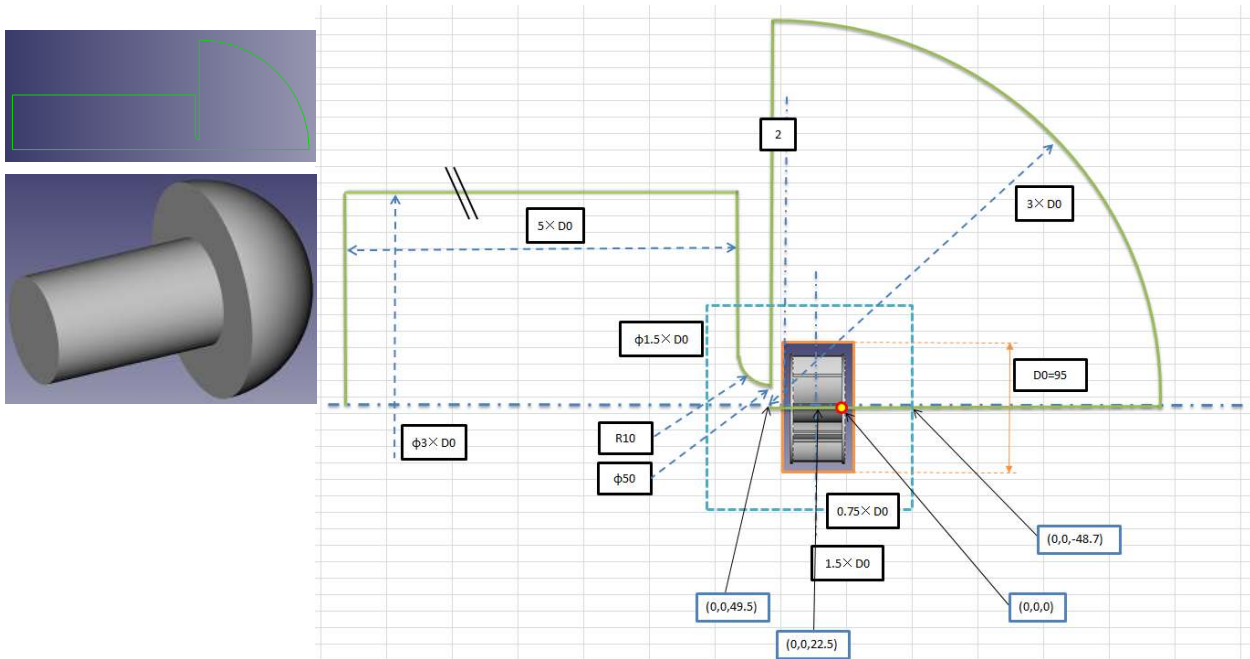
## 2. モデルの作成 チャンバー、回転領域等

2013.6.15

20/83



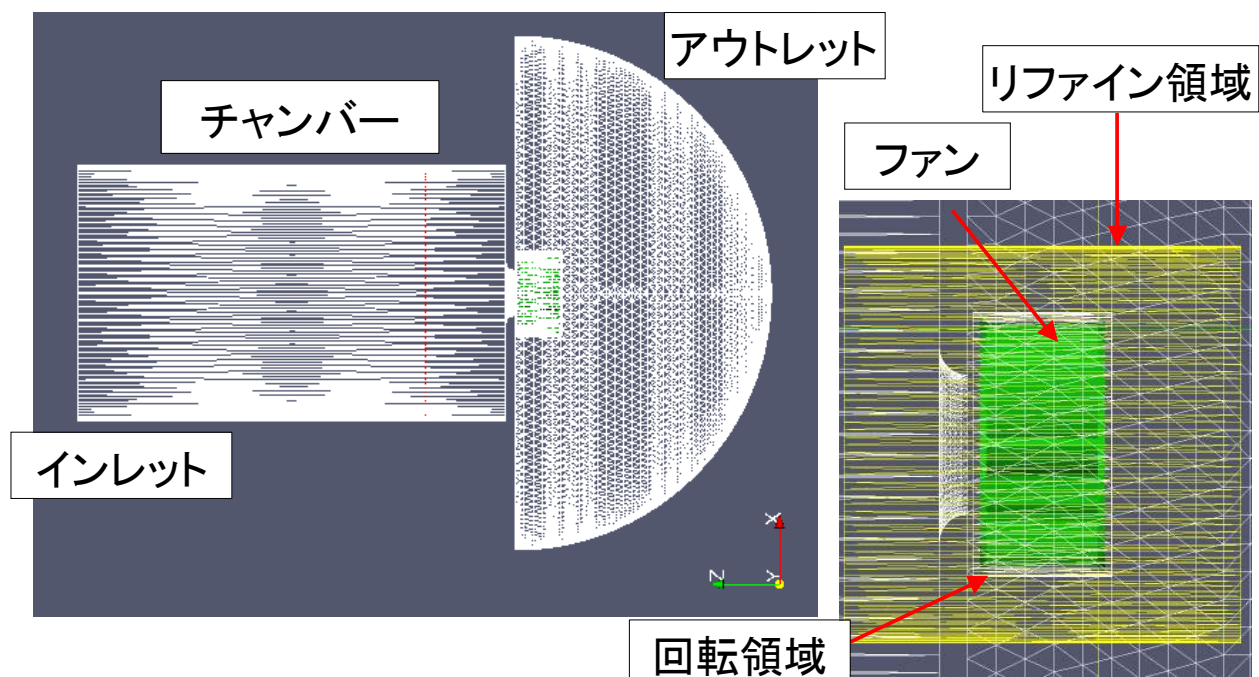
2. モデルの作成 チャンバー, 回転領域等  
同様な方法で作図, stl化する



2013.6.15

21/83

2. モデルの作成 チャンバー, 回転領域等



2013.6.15

22/83

## 3. MRFでの計算

2013.6.15

23/83

### 3-1. メッシュの作成 MRF

2013.6.15

24/83

## 3-1. メッシュの作成 (MRF)

## バッチ処理で実施

- |                        |              |
|------------------------|--------------|
| ①計算に必要なファイルの設定         |              |
| ②blockMesh             | 基礎メッシュの作成    |
| ③surfaceFeatureExtract | 特徴線の抽出       |
| ④decomposePar          | 並列メッシュの準備    |
| ⑤snappyHexMesh         | 並列メッシュの作成    |
| ⑥reconstructParMesh    | メッシュの統合      |
| ⑦checkMesh             | チェックのメッシュ    |
| ⑧renumberMesh          | メッシュのリナンバリング |
| ⑨ホルダ構成の設定・調整           |              |

2013.6.15

25/83

## 3-1. メッシュの作成 (MRF)

```

echo "Start mesh MRF"
CASE_DIR=case_set/case1/set
BLOCK_MESH=blockMeshDict_case1
DECOMPOSE_PAR=decomposeParDict_case1_mesh
SNAPPYHEXMESH_DICT=snappyHexMeshDict_case1
SNAPPYHEXMESH_FEATURE=surfaceFeatureExtractDict_1
SNAPPYHEXMESH_DEFAULT=surfaceFeatureExtractDictDefaults_1
CONTROL_DICT=controlDict_case1_mesh

```

①計算に必要なファイルの設定

```

echo "blockMesh"
cp -r $CASE_DIR/$CONTROL_DICT system/controlDict
cp -r $CASE_DIR/$BLOCK_MESH constant/polyMesh/blockMeshDict
$runApplication blockMesh > log.blockMesh

```

②基礎メッシュの作成

```

echo "surfaceFeatureExtract"
cp -r $CASE_DIR/$SNAPPYHEXMESH_FEATURE system/surfaceFeatureExtractDict
cp -r $CASE_DIR/$SNAPPYHEXMESH_DEFAULT system/surfaceFeatureExtractDictDefaults
$runApplication surfaceFeatureExtract

```

③特徴線の抽出

```

echo "decomposePar 4 blocks"
cp -r $CASE_DIR/$DECOMPOSE_PAR system/decomposeParDict
$runApplication decomposePar > log.decomposePar

```

④並列メッシュの準備

```

echo "Copy data"
cp -r constant/triSurface/ processor0/constant/
cp -r constant/triSurface/ processor1/constant/
cp -r constant/triSurface/ processor2/constant/
cp -r constant/triSurface/ processor3/constant/

```

2013.6.15

26/83

### 3-1. メッシュの作成(MRF)

```

echo "snappyHexMesh 4CPU"
cp -r $CASE_DIR/$SNAPPYHEXMESH_DICT system/snappyHexMeshDict
$runApplication mpirun -np 4 snappyHexMesh -parallel > log.snappyHexMesh

echo "reconstructParMesh"
$runApplication reconstructParMesh -time 3 -mergeTol 1e-6 > log.reconstructParMesh

rm -r processor0
rm -r processor1
rm -r processor2
rm -r processor3

echo "checkMesh"
$runApplication checkMesh > log.checkMesh

echo "renumberMesh"
$runApplication renumberMesh -time 3 > log.renumberMesh

echo "rename holder name"
mv constant/polyMesh constant/polyMesh_blockMesh

echo "set data"
mv 4/polyMesh constant
rm -r 2
rm -r 3
rm -r 4

echo "End of mesh"
    
```

⑤並列メッシュの作成

⑥メッシュの統合

⑦チェックメッシュ

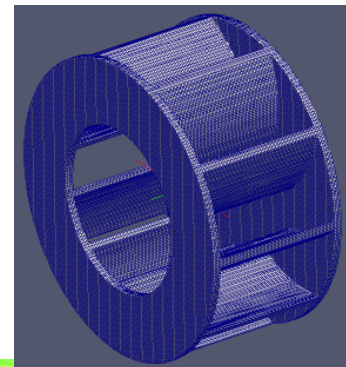
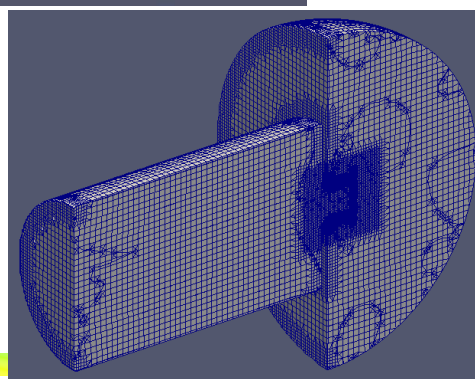
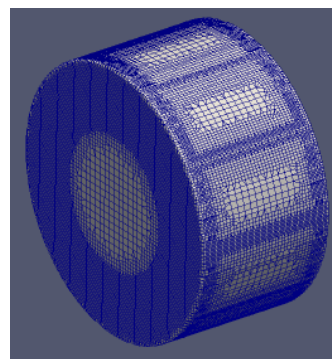
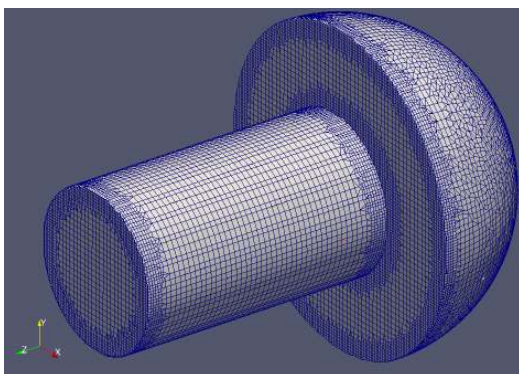
⑧メッシュのリナンバリング

⑨ホルダ構成の設定・調整

2013.6.15

27/83

### 3-1. メッシュの作成(MRF)



2013.6.15

28/83



### 3-1. メッシュの作成(MRF)

```

Checking geometry...
Overall domain bounding box (-0.284989 -0.284982 -0.235512) (0.284989 0.285007 0.534657)
Mesh (non-empty, non-wedge) directions (1 1 1)
Mesh (non-empty) directions (1 1 1)
Boundary openness (-2.59726e-16 -7.59549e-16 -6.60444e-15) OK.
Max cell openness = 5.86112e-16 OK.
Max aspect ratio = 22.3317 OK.
Minimum face area = 1.19564e-08. Maximum face area = 0.000177886. Face area magnitudes OK.
Min volume = 3.92571e-12. Max volume = 1.29464e-06. Total volume = 0.078642. Cell volumes OK.
Mesh non-orthogonality Max: 65.7181 average: 8.79805
Non-orthogonality check OK.
Face pyramids OK.
Max skewness = 1.7374 OK.
Coupled point location match (average 0) OK.
    
```

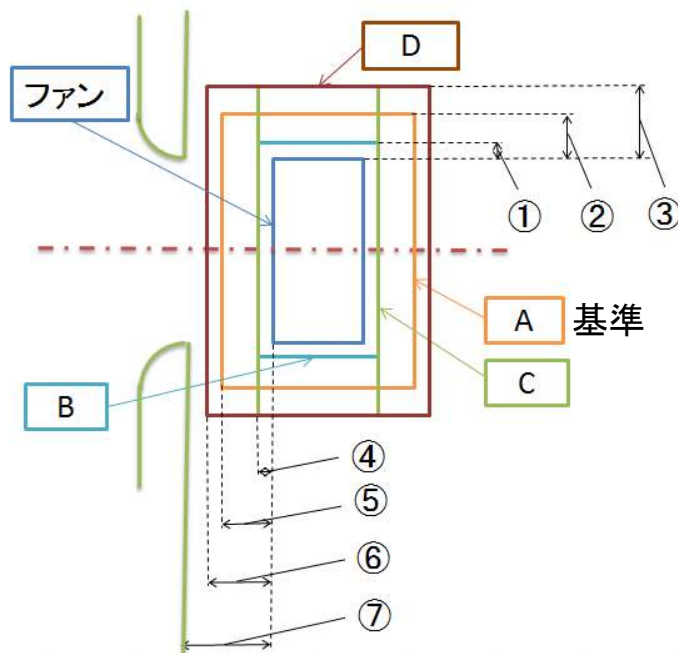
Mesh OK.

今回のMRF計算で利用したメッシュの品質

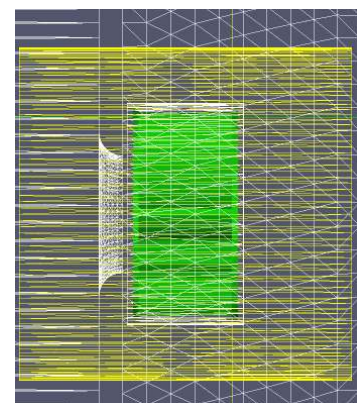
2013.6.15

29/83

### 3-1. メッシュの作成(MRF) 回転領域の寸法設定 4パターン



位置	寸法
①	1
②	2.5
③	4
④	1
⑤	2.5
⑥	3
⑦	4.5

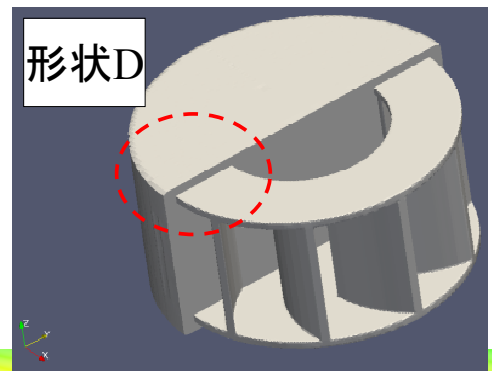
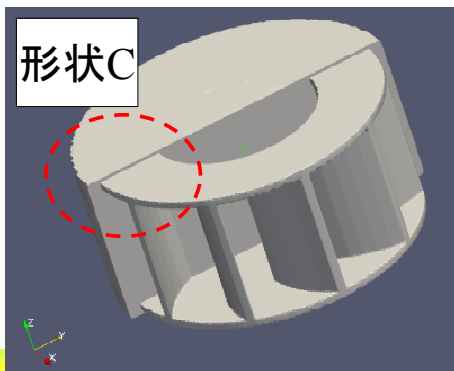
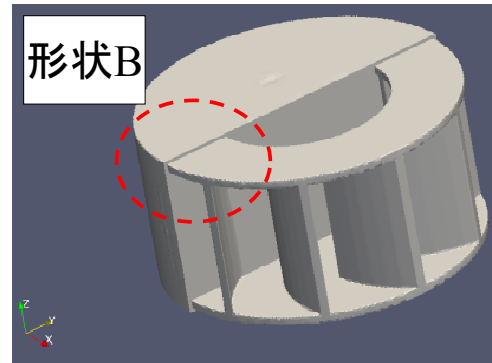
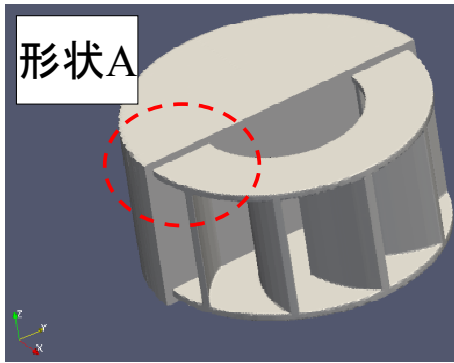


2013.6.15

30/83

### 3-1. メッシュの作成 (MRF)

基準



2013.6.15

31/83

## 3-2. 計算の設定 MRF

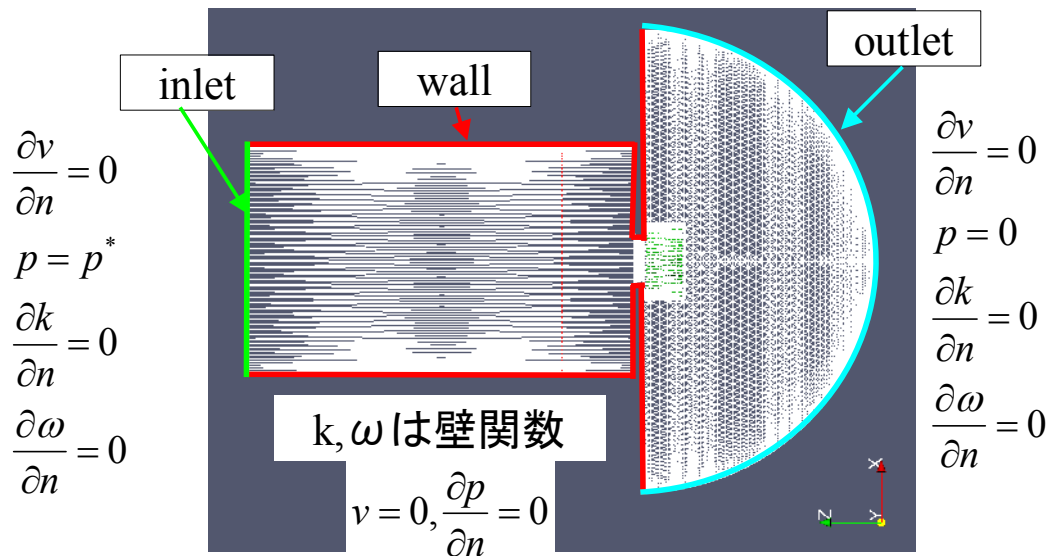
2013.6.15

32/83



### 3-2. 計算の設定 (MRF)

#### 境界条件



P\*: 明示的に設定する

### 3-2. 計算の設定 (MRF)

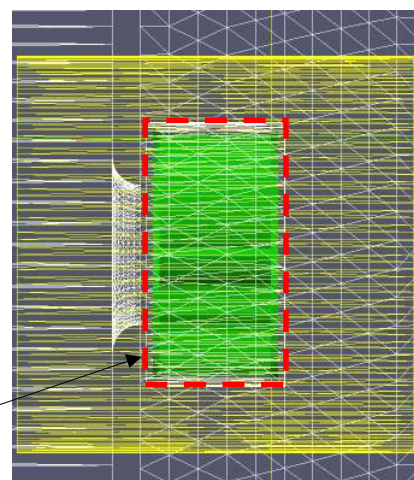
回転部分の設定は, system/fvOptionsで行う。

```

MRF1
{
  type          MRFSource;
  active        true;
  selectionMode cellZone;
  cellZone     rotate_area;

  MRFSourceCoeffs
  {
    origin      (0 0 0);
    axis        (0 0 -1);
    omega       314; //3000rpm
  }
}
    
```

回転領域  
3000rpm



ファンの周速度 14.1m/s

### 3-2. 計算の設定(MRF)

fvSchemes/divSchemes, fvSolution/relaxationFactorsは3設定  
(その他の部分は同じ)

設定1	設定2	設定3
<pre>div(phi,U) bounded Gauss upwind; div(phi,k) bounded Gauss upwind; div(phi,epsilon) bounded Gauss upwind; div(phi,omega) bounded Gauss upwind;</pre>	<pre>div(phi,U) bounded Gauss upwind; div(phi,k) bounded Gauss upwind; div(phi,epsilon) bounded Gauss upwind; div(phi,omega) bounded Gauss upwind;</pre>	<pre>div(phi,U) bounded Gauss limitedLinearV 1; div(phi,k) bounded Gauss limitedLinear 1; div(phi,epsilon) bounded Gauss limitedLinear 1; div(phi,omega) bounded Gauss limitedLinear 1;</pre>
<pre>relaxationFactors {   fields   {     p 0.2;   }   equations   {     u 0.5;     k 0.5;     omega 0.5;   } }</pre>	<pre>relaxationFactors {   fields   {     p 0.3;   }   equations   {     u 0.7;     k 0.7;     omega 0.7;   } }</pre>	<pre>relaxationFactors {   fields   {     p 0.2;   }   equations   {     u 0.5;     k 0.5;     omega 0.5;   } }</pre>

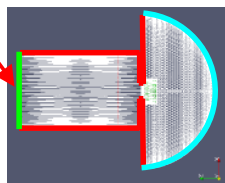
緩和係数の違い      スキームの差

### 3-2. 計算の設定(MRF)

PQ特性の計算では、チャンバー側の $p^*$ を規定して計算を行う。

$\Delta p(\text{Pa})$	$\Delta p/\rho$	$\phi(\text{m}^3/\text{s})$	$\phi(\text{m}^3/\text{min})$	流量係数	静圧係数	$Mz(p)$	$Mz(\nu)$	回転負荷(W)	軸動力係数
0	0.000	-0.020	1.174	0.132	0.000	0.006	0.000	2.123	0.119
5	4.149	-0.019	1.161	0.130	0.042	0.006	0.000	2.152	0.120
10	8.299	-0.019	1.132	0.127	0.083	0.007	0.000	2.192	0.123
15	12.448	-0.019	1.116	0.125	0.125	0.007	0.000	2.238	0.125
20	16.598	-0.018	1.090	0.122	0.166	0.007	0.000	2.229	0.125
25	20.747	-0.018	1.058	0.119	0.208	0.007	0.000	2.262	0.126
30	24.896	-0.017	1.037	0.116	0.249	0.007	0.000	2.369	0.132
35	29.046	-0.017	1.018	0.114	0.291	0.007	0.000	2.363	0.132
40	33.195	-0.016	0.975	0.109	0.332	0.007	0.000	2.378	0.133
45	37.344	-0.015	0.929	0.104	0.374	0.007	0.000	2.373	0.133
50	41.494	-0.015	0.876	0.098	0.415	0.007	0.000	2.355	0.132
60	49.793	-0.013	0.766	0.086	0.498	0.007	0.000	2.288	0.128
70	58.091	-0.011	0.651	0.073	0.581	0.007	0.000	2.190	0.122
80	66.390	-0.008	0.484	0.054	0.664	0.006	0.000	1.961	0.110
90	74.689	-0.003	0.183	0.020	0.747	0.004	0.000	1.412	0.079

負圧で設定



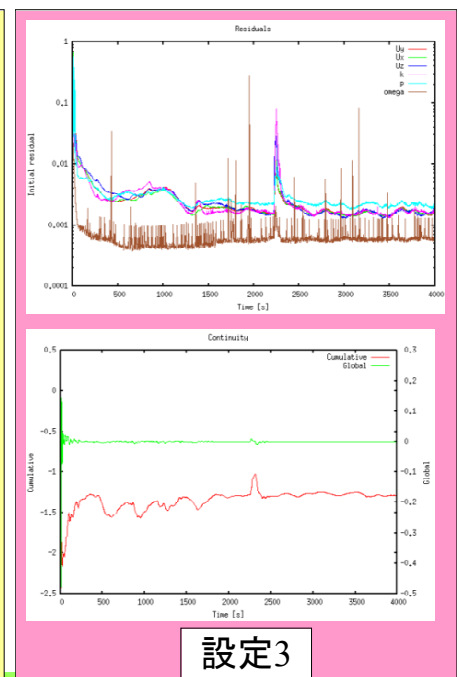
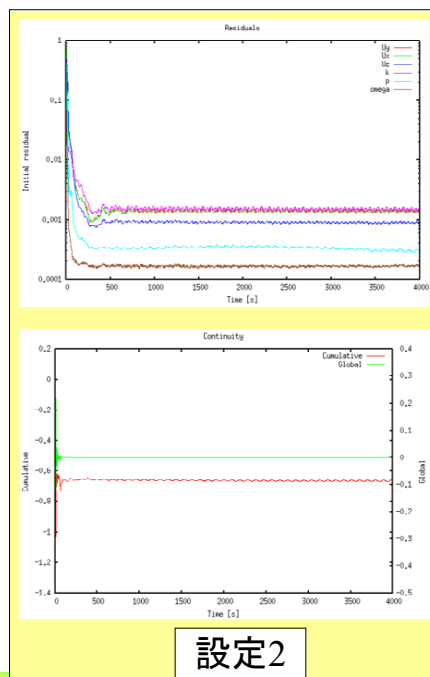
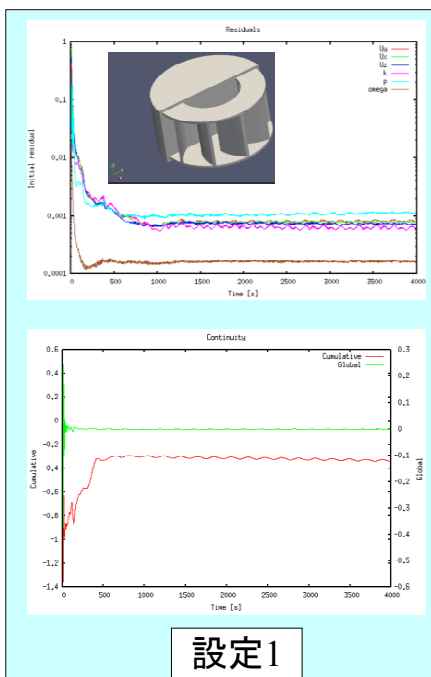
# 3-3. 計算の結果 MRF

2013.6.15

37/83

## 3-3. 計算の結果 (MRF) 収束状態 (形状モデルはA)

設定	fvSchemes/divSchemes	fvSolutions/relaxationFactors			
		p	U	k	omega
1	bounded Gauss upwind	0.2	0.5	0.5	0.5
2	bounded Gauss upwind	0.3	0.7	0.7	0.7
3	bounded Gauss limitedLinear 1	0.2	0.5	0.5	0.5



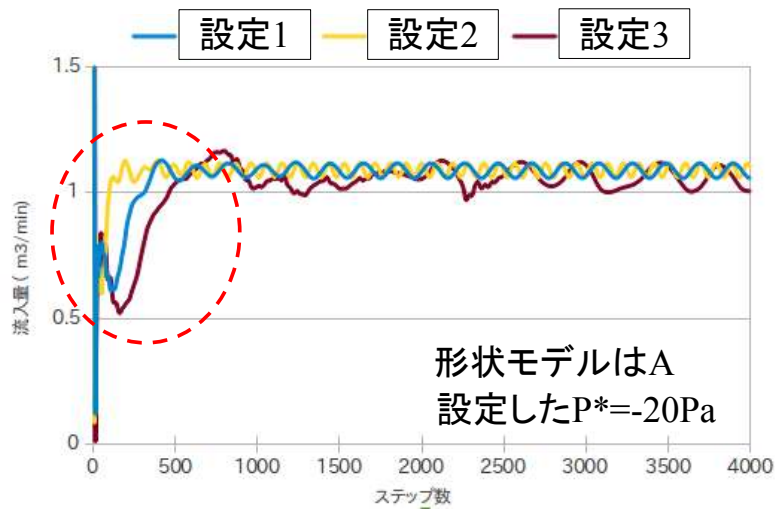
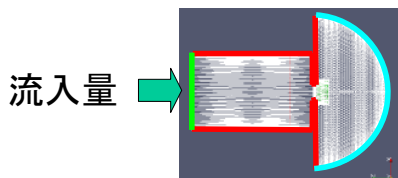
2013.6.15

設定したP\*=-20Pa

38/83

### 3-3. 計算の結果 (MRF)

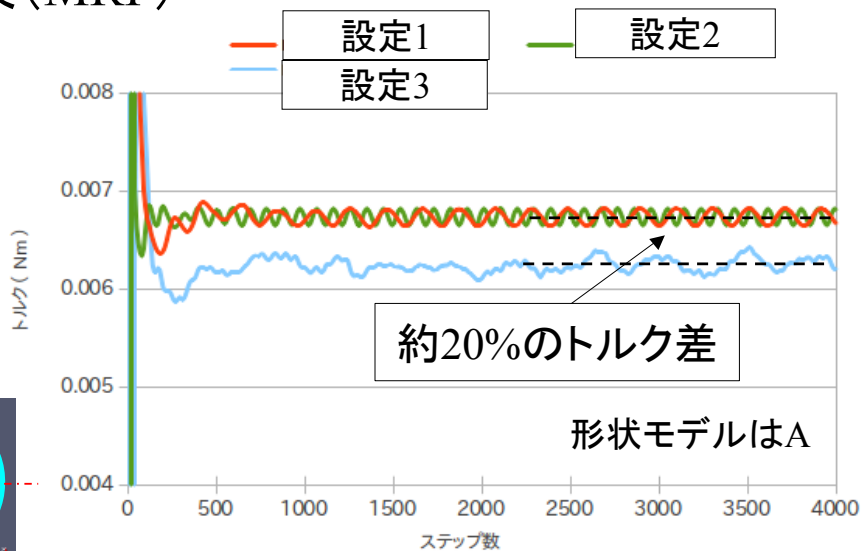
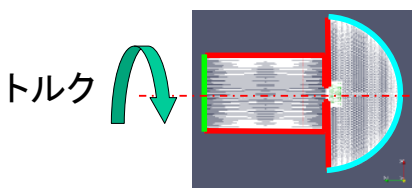
- ・緩和係数が大きいと初期の収束が速くなる。
- ・upwindに比較してlimitedLinearは収束、安定に回数が必要。
- ・limitedLinearはupwindより少し流入量が少ない。



設定	fvSchemes/divSchemes	fvSolutions/relaxationFactors				流入量	
		p	U	k	omega	m <sup>3</sup> /min	比率
1	bounded Gauss upwind	0.2	0.5	0.5	0.5	1.089	100
2	bounded Gauss upwind	0.3	0.7	0.7	0.7	1.090	100
3	bounded Gauss limitedLinear 1	0.2	0.5	0.5	0.5	1.061	97

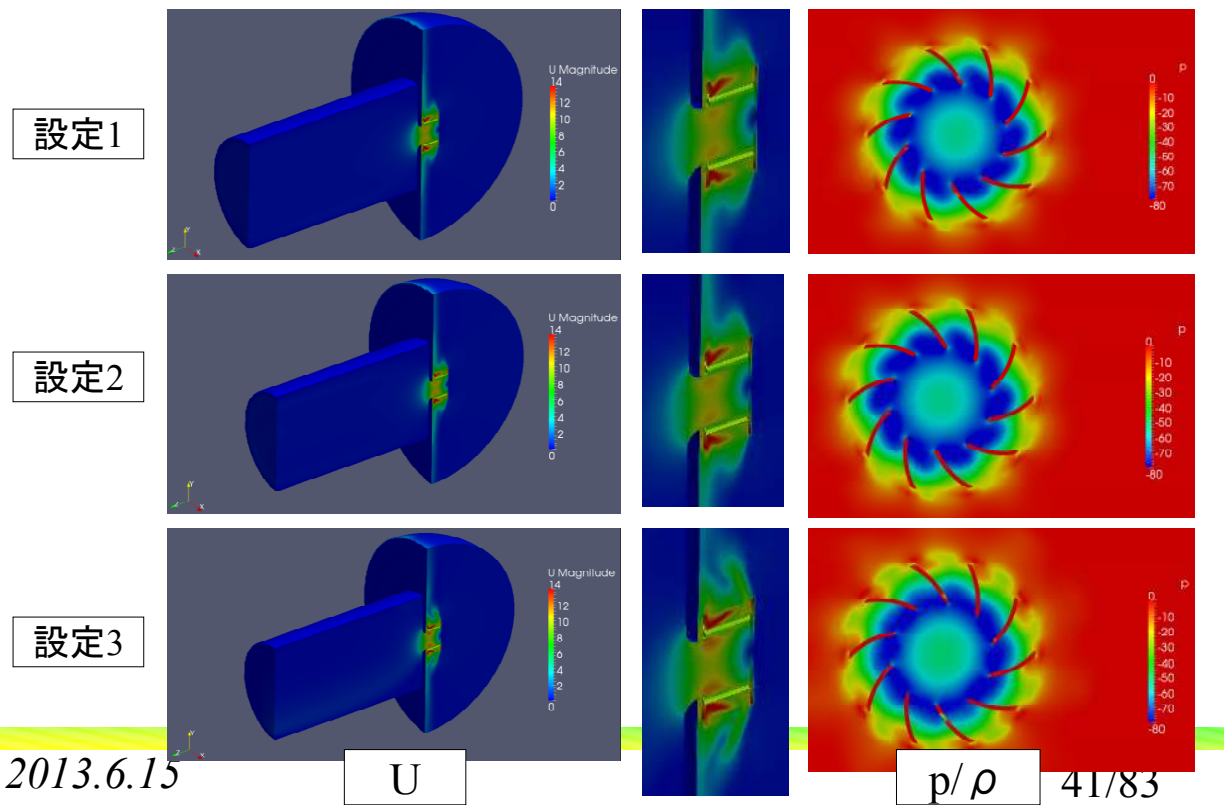
### 3-3. 計算の結果 (MRF)

- ・upwindとlimitedLinearでは、トルクの差がある。解法の差による数値粘性が影響している？



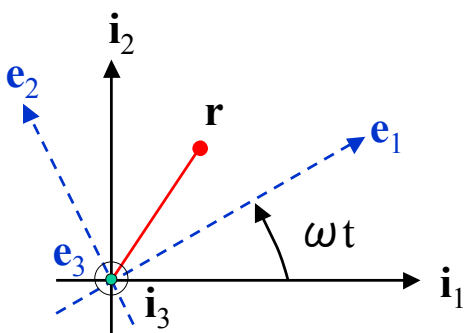
設定	fvSchemes/divSchemes	fvSolutions/relaxationFactors				トルク	
		p	U	k	omega	N·cm	比率
1	bounded Gauss upwind	0.2	0.5	0.5	0.5	0.684	100
2	bounded Gauss upwind	0.3	0.7	0.7	0.7	0.685	100
3	bounded Gauss limitedLinear 1	0.2	0.5	0.5	0.5	0.563	82

### 3-3. 計算の結果(MRF)



### 3-3. 計算の結果(MRF)

ファンの計算では、ファンから見た流れを見たい場合がある。絶対速度でなく、回転座標系での速度となる。簡単のため、z軸回り(i<sub>3</sub>軸回り)に回転する座標系を考えてみる。



$$r = xi_1 + yi_2 + zi_3 = Xe_1 + Ye_2 + Ze_3$$

$$e_1(t) = \cos(\omega t)i_1 + \sin(\omega t)i_2$$

$$e_2(t) = -\sin(\omega t)i_1 + \cos(\omega t)i_2$$

$$e_3(t) = i_3$$

$$\frac{dr}{dt} = \frac{dx}{dt}i_1 + \frac{dy}{dt}i_2 + \frac{dz}{dt}i_3$$

$$= \frac{dX}{dt}e_1 + \frac{dY}{dt}e_2 + \frac{dZ}{dt}e_3 + X \frac{de_1}{dt} + Y \frac{de_2}{dt} + Z \frac{de_3}{dt}$$

$$= \left( \frac{dX}{dt} - \omega Y \right) e_1 + \left( \frac{dY}{dt} + \omega X \right) e_2 + \frac{dZ}{dt} e_3$$

### 3-3. 計算の結果 (MRF)

この式をparaView上で見るには、次の方法があるようで、解説があります。

- ・公開されているユーティリティを使う。

(<http://www.cfd-online.com/Forums/openfoam-solving/71277-how-derive-relative-velocity-mrfsimplefoam.html#post350891>)

- ・ParaViewのCalculatorを使う。

(<https://groups.google.com/forum/?fromgroups#!topic/openfoam/WrXQmo39HrQ>)

2013.6.15

43/83

### 3-3. 計算の結果 (MRF)

- ・公開されているユーティリティを使う。

まずは2つのファイルを持ってくる。

The screenshot shows a forum post from user 'linnemann' dated May 16, 2013. The post discusses MRF properties for OF-2.2.x. It includes two edit messages: the first explains that MRF has changed in 2.2 and some solvers use MRFPProperties in the constant folder, while others use system/fvOptions; the second instructs to place the MRFPProperties file in the constant folder with the same info as the one from system/fvOptions. Below the text are two attached files: 'Urel-2.2.x.tar.gz (1.4 KB, 5 views)' and 'MRFPProperties.txt (1.1 KB, 6 views)'. A red dashed circle highlights these two files, with arrows pointing from the labels 'ユーティリティ' (pointing to the tar.gz file) and 'constantに入れる設定ファイル' (pointing to the txt file).

constantに入れる設定ファイル

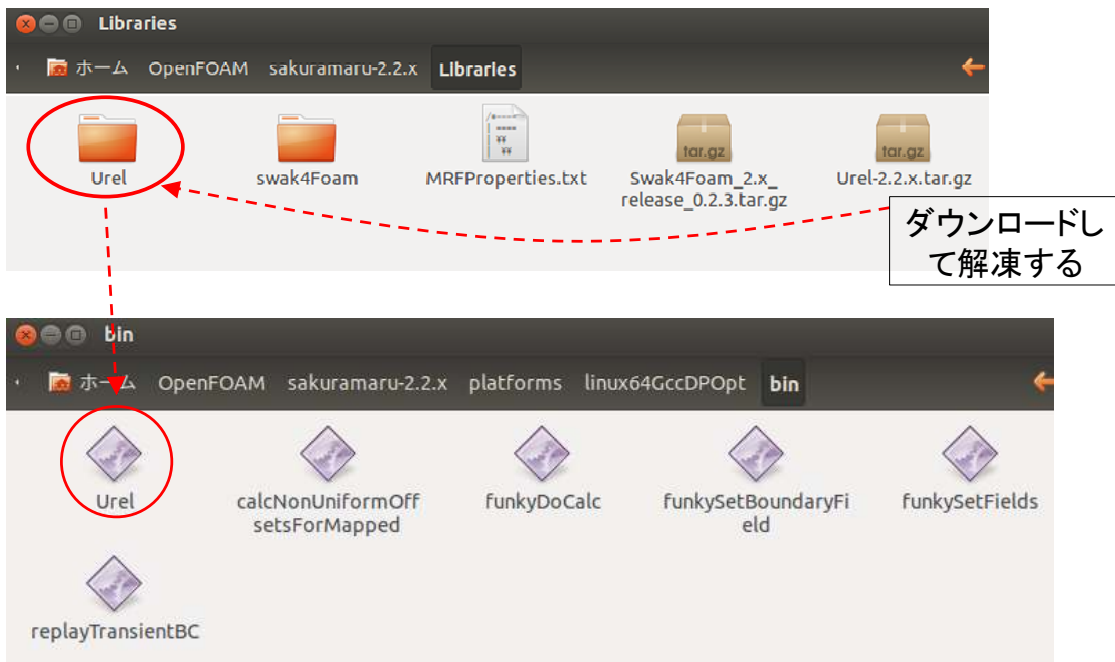
2013.6.15

44/83



### 3-3. 計算の結果(MRF)

ファイルを解凍してwmakeする。Urelが作成される。



2013.6.15

45/83

### 3-3. 計算の結果(MRF)

コンパイルが出来たか確認する。

```
sakuramaru@SAKURA-MARU:~/Desktop/ver_2.2.x/small_fan/MRF/MRF_checks$ Urel -help
Usage: Urel [OPTIONS]
options:
  -case <dir>          specify alternate case directory, default is the cwd
  -constant            include the 'constant/' dir in the times list
  -latestTime         select the latest time
  -noFunctionObjects  do not execute functionObjects
  -noZero             exclude the '0/' dir from the times list, has precedence
                    over the -zeroTime option
  -parallel           run in parallel
  -roots <dir1 .. dirN>
                    slave root directories for distributed running
  -time <ranges>     comma-separated time ranges - eg, ':10,20,40:70,1000:'
  -srcDoc             display source code in browser
  -doc               display application documentation in browser
  -help              print the usage

Using: OpenFOAM-2.2.x (see www.OpenFOAM.org)
Build: 2.2.x-ff9ffb9699bd
```

2013.6.15

46/83

### 3-3. 計算の結果 (MRF)

#### Urelの実施

constantにMRFPropertiesを設定する。

```

MRFProperties *
-----
Field Operation And Manipulation
OpenFOAM: The Open Source CFD Toolbox
Version: 2.2.0
Web: www.OpenFOAM.org

FoamFile
{
  version      2.0;
  format       ascii;
  class        dictionary;
  location     "constant";
  object       MRFProperties;
}
// *****

MRF1
{
  cellZone     rotate_area;
  active       yes;

  // Fixed patches (by default they 'move' with the MRF zone)
  nonRotatingPatches ();

  origin       (0 0 0);
  axis         (0 0 -1);
  omega        314; //3000rpm
}
// *****
    
```

#### fvOptions

```

MRF1
{
  type         MRFSource;
  active       true;
  selectionMode cellZone;
  cellZone     rotate_area;

  MRFSourceCoeffs
  {
    origin      (0 0 0);
    axis        (0 0 -1);
    omega       314; //3000rpm
  }
}
    
```



### 3-3. 計算の結果 (MRF)

#### Urelの実施

```

sakuramaru@SAKURA-MARU:~/Desktop/ver_2.2.x/small_fan/MRF/MRF_checks$ Urel -latestTime
-----
Field Operation And Manipulation
OpenFOAM: The Open Source CFD Toolbox
Version: 2.2.x
Web: www.OpenFOAM.org
-----

Build : 2.2.x-ff9ffb9699bd
Exec   : Urel -latestTime
Date   : Jun 09 2013
Time   : 08:36:54
Host   : "SAKURA-MARU"
PID    : 11749
Case   : /home/sakuramaru/Desktop/ver_2.2.x/small_fan/MRF/MRF_check
nProcs : 1
sigFpe : Enabling floating point exception trapping (FOAM_SIGFPE).
fileModificationChecking : Monitoring run-time modified files using timeStampMaster
allowSystemOperations : Disallowing user-supplied system call operations

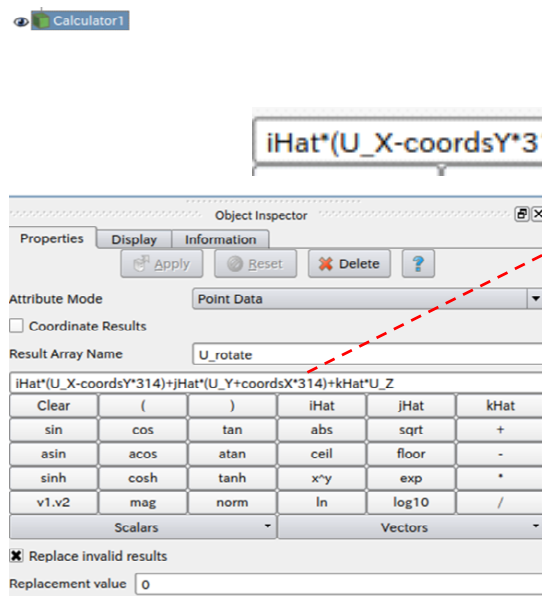
// ***** //
Create time

Create mesh for time = 4000

Creating MRF zone list from MRFProperties
  creating MRF zone: MRF1
Time = 4000
  Reading U
  Calculating Urel
    
```

### 3-3. 計算の結果 (MRF)

- ・ParaViewのCalculatorを使う。



$$i\hat{H}at*(U\_X-coordsY*314)+j\hat{H}at*(U\_Y+coordsX*314)+k\hat{H}at*U\_Z$$

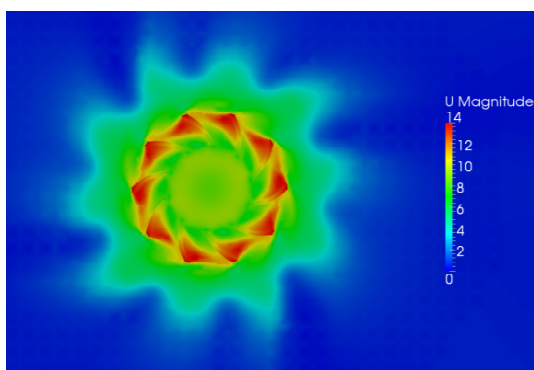
この式の説明は下記を参照  
<https://groups.google.com/forum/?fromgroups#!topic/openfoam/WrXQmo39HrQ>

2013.6.15

49/83

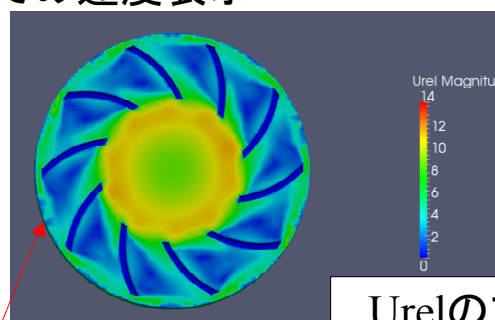
### 3-3. 計算の結果 (MRF)

- ・2つの方法による回転座標系での速度表示

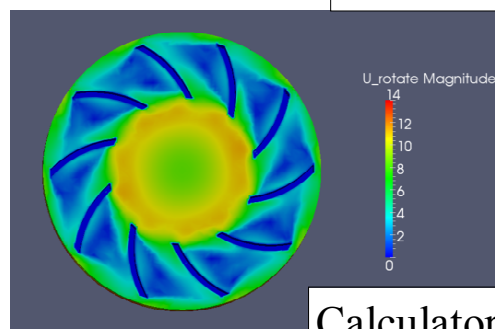


全体座標系での速度

Urelの方法では外周部分で速度のみだれがある。



Urelの方法



Calculatorの方法

2013.6.15

50/83

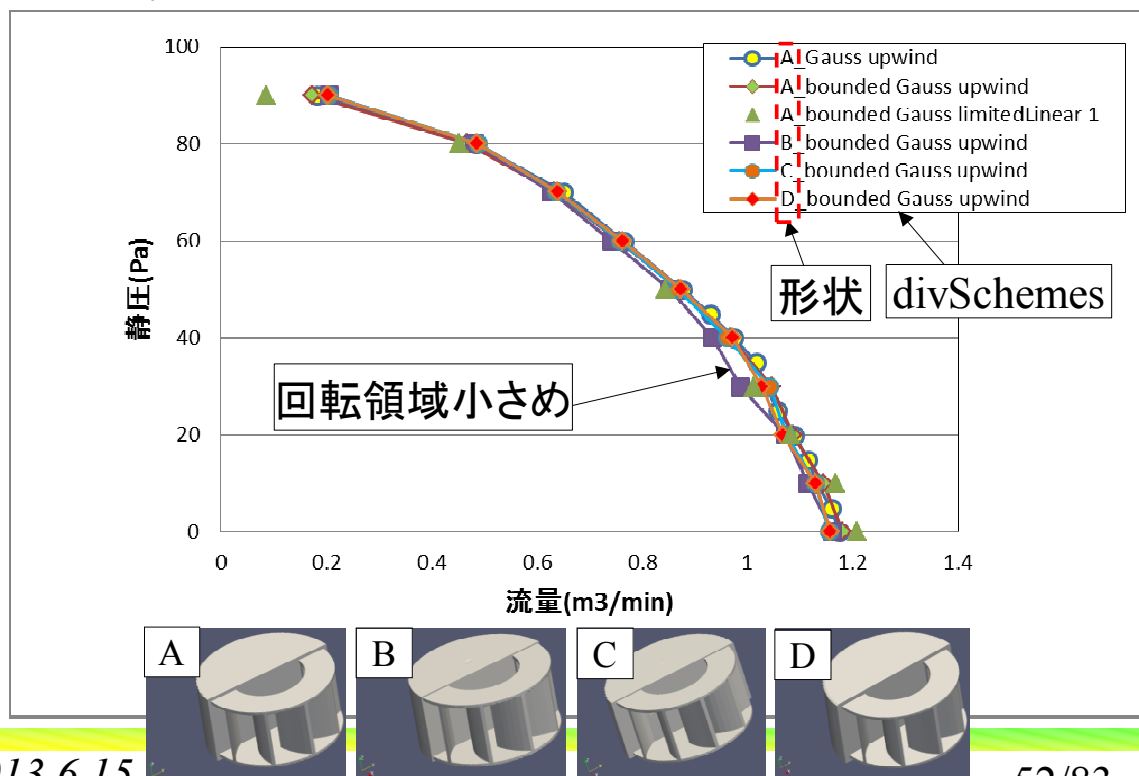
# 3-4. 特性計算の結果

## MRF

2013.6.15

51/83

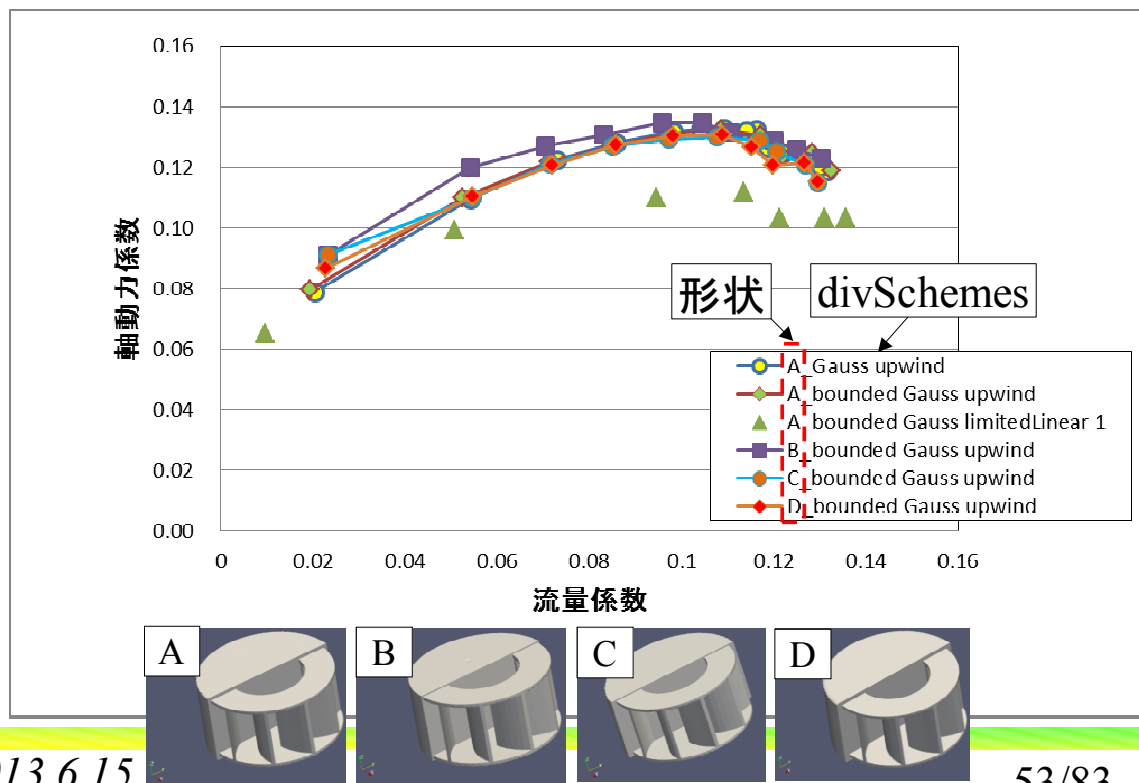
### 3-4. PQ特性の結果(MRF)



2013.6.15

52/83

### 3-4. 無次元特性の結果 (MRF)



2013.6.15

53/83

## 4. AMIでの計算

2013.6.15

54/83

# 4-1. メッシュの作成 AMI

2013.6.15

55/83

## 4-1. メッシュの作成 (AMI)

- |                        |            |
|------------------------|------------|
| ①計算に必要なファイルの設定         |            |
| ②blockMesh             | 基礎メッシュの作成  |
| ③surfaceFeatureExtract | 特徴線の抽出     |
| ④decomposePar          | 並列メッシュの準備  |
| ⑤snappyHexMesh         | 並列メッシュの作成  |
| ⑥reconstructParMesh    | メッシュの統合    |
| ⑦topoSet               | ダミーフェースの作成 |
| ⑧createPatch           | パッチの作成     |
| ⑨createBaffles         | 回転面の設定     |
| ⑩mergeOrSplitBaffles   | 回転面の分割     |
| ⑪checkMesh             | チェックのメッシュ  |
| ⑫ホルダ構成の設定・調整           |            |

2013.6.15

56/83



## 4-1. メッシュの作成 (AMI)

```
echo "Start mesh AMI"
CASE_DIR=case_set/case1/set
BLOCK_MESH=blockMeshDict_case1
DECOMPOSE_PAR=decomposeParDict_case1_mesh
SNAPPYHEXMESH_DICT=snappyHexMeshDict_case1
FV_SCH=fvSchemes_case1
FV_SOL=fvSolution_case1
CONTROL_DICT=controlDict_case1_mesh
DummyfaceSet=DummyfaceSet.topoSetDict_case1
CREATE_PatchDict=createPatchDict_case1
CREATE_BafflesDict=createBafflesDict_case1
AMI_AREA=rotate_area
cp -r $CASE_DIR/$FV_SCH system/fvSchemes
cp -r $CASE_DIR/$FV_SOL system/fvSolution
```

①計算に必要なファイルの設定

```
echo "blockMesh"
cp -r $CASE_DIR/$CONTROL_DICT system/controlDict
cp -r $CASE_DIR/$BLOCK_MESH constant/polyMesh/blockMeshDict
$runApplication blockMesh > log.blockMesh
```

②基礎メッシュの作成

```
echo "surfaceFeatureExtract"
$runApplication surfaceFeatureExtract
```

③特徴線の抽出し

```
echo "decomposePar 4 blocks"
cp -r $CASE_DIR/$DECOMPOSE_PAR system/decomposeParDict
$runApplication decomposePar > log.decomposePar
```

④並列メッシュの準備

2013.6.15

57/83

## 4-1. メッシュの作成 (AMI)

```
echo "Copy data"
cp -r constant/triSurface/ processor0/constant/
cp -r constant/triSurface/ processor1/constant/
cp -r constant/triSurface/ processor2/constant/
cp -r constant/triSurface/ processor3/constant/
```

```
echo "snappyHexMesh 4CPU"
cp -r $CASE_DIR/$SNAPPYHEXMESH_DICT system/snappyHexMeshDict
$runApplication mpirun -np 4 snappyHexMesh -parallel > log.snappyHexMesh
```

⑤並列メッシュの作成

```
echo "reconstructParMesh"
$runApplication reconstructParMesh -time 3 -mergeTol 1e-6 > log.reconstructParMesh
rm -r processor0
rm -r processor1
rm -r processor2
rm -r processor3
```

⑥メッシュの統合

```
echo "topoSet DummyfaceSet"
cp -r $CASE_DIR/$DummyfaceSet system/topoSetDict
$runApplication topoSet > log.dummyfaceset
```

⑦ダミーフェースの作成

```
echo "createPatch"
cp -r $CASE_DIR/$CREATE_PatchDict system/createPatchDict
$runApplication createPatch > log.createPatch
```

⑧パッチの作成

```
echo "createBaffles"
cp -r $CASE_DIR/$CREATE_BafflesDict system/createBafflesDict
$runApplication createBaffles > log.createBaffles
```

⑨回転面の設定

2013.6.15

58/83

## 4-1. メッシュの作成 (AMI)

```

echo "mergeOrSplitBaffles"
$runApplication mergeOrSplitBaffles -split > log.mergeOrSplitBaffles

```

⑩回転面の分割

```

echo "checkMesh"
$runApplication checkMesh > log.checkMesh

```

⑪チェックメッシュ

```

echo "rename holder name"
mv constant/polyMesh constant/polyMesh_blockMesh

```

⑫ホルダ構成の設定・調整

```

echo "set data"
mv 5/polyMesh constant

echo "clear holder"
rm -r 1
rm -r 2
rm -r 3
rm -r 4
rm -r 5

echo "End of mesh"

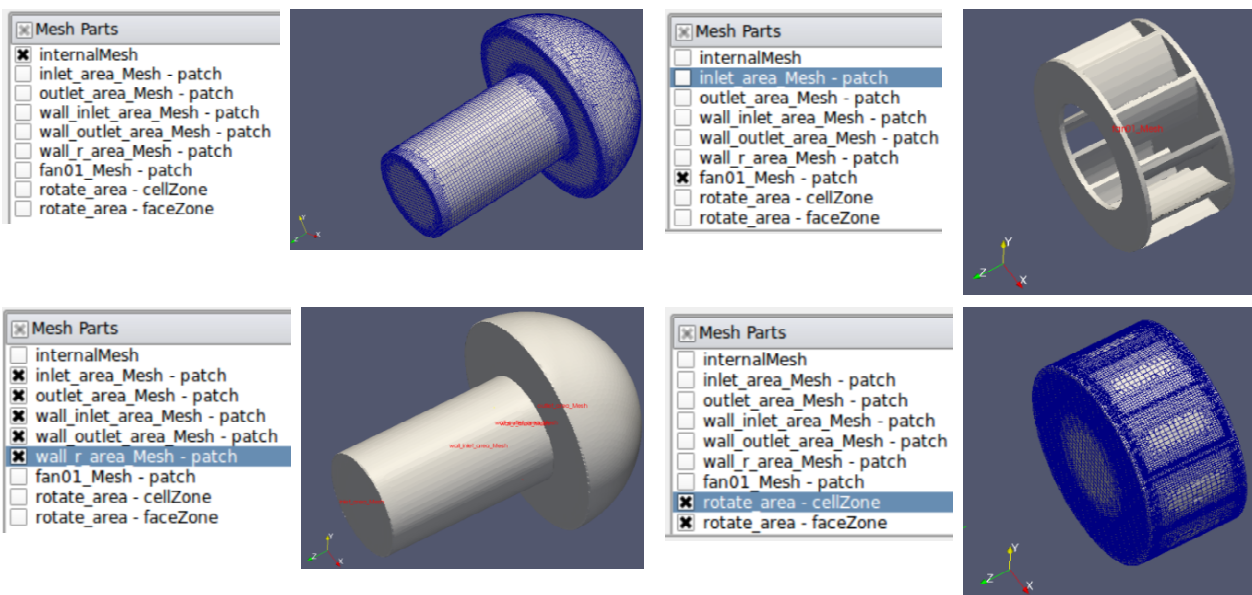
```

2013.6.15

59/83

## 4-1. メッシュの作成 (AMI)

### ⑤snappyHexMesh 並列メッシュの作成



2013.6.15

60/83

## 4-1. メッシュの作成 (AMI)

### ⑦ topoSet ダミーフェースの作成

```

sakuramaru@SAKURA-MARU:~/Desktop/ver_2.2.x/small_fan_AMI$ topoSet
-----
Field      | OpenFOAM: The Open Source CFD Toolbox
Operation  | Version: 2.2.x
And        | Web: www.OpenFOAM.org
Manipulation
-----
Build   : 2.2.x-b477bcabd47f
Exec    : topoSet
Date    : May 11 2013
Time    : 18:21:24
Host    : "SAKURA-MARU"
PID     : 17914
Case    : /home/sakuramaru/Desktop/ver_2.2.x/small_fan_AMI
nProcs  : 1
sigFpe  : Enabling floating point exception trapping (FOAM_SIGFPE).
fileModificationChecking : Monitoring run-time modified files using timeStampMaster
allowSystemOperations : Disallowing user-supplied system call operations

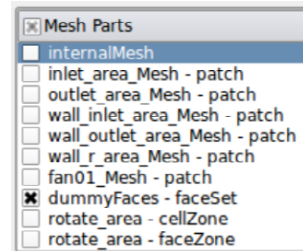
// *****
Create time

Create polyMesh for time = 2

Reading topoSetDict

Time = 2
mesh not changed.
Created faceSet dummyFaces
Applying source labelToFace
Adding faces mentioned in dictionary ...
faceSet dummyFaces now size 0

End
    
```



dummyFacesは形として作成されない。paraViewで表示なし

2013.6.15

61/83

## 4-1. メッシュの作成 (AMI)

### ⑧ createPatch パッチの作成

```

sakuramaru@SAKURA-MARU:~/Desktop/ver_2.2.x/small_fan_AMI$ createPatch
-----
Field      | OpenFOAM: The Open Source CFD Toolbox
Operation  | Version: 2.2.x
And        | Web: www.OpenFOAM.org
Manipulation
-----
Build   : 2.2.x-b477bcabd47f
Exec    : createPatch
Date    : May 11 2013
Time    : 18:28:30
Host    : "SAKURA-MARU"
PID     : 17927
Case    : /home/sakuramaru/Desktop/ver_2.2.x/small_fan_AMI
nProcs  : 1
sigFpe  : Enabling floating point exception trapping (FOAM_SIGFPE).
fileModificationChecking : Monitoring run-time modified files using timeStampMaster
allowSystemOperations : Disallowing user-supplied system call operations

// *****
Create time

Create polyMesh for time = 2

Reading createPatchDict

Adding new patch AMI1 as patch 9 from
{
  type          cyclicAMI;
  neighbourPatch AMI2;
  transform     noOrdering;
  surface
  {
  }
}

Adding new patch AMI2 as patch 10 from
{
  type          cyclicAMI;
  neighbourPatch AMI1;
  transform     noOrdering;
  surface
  {
  }
}
    
```

```

Read 0 faces from faceSet dummyFaces
Read 0 faces from faceSet dummyFaces

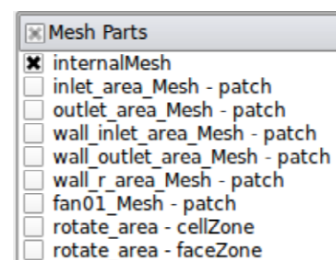
Doing topology modification to order faces.

AMI: Creating addressing and weights between 0 source faces and 0 target faces
Not synchronising points.

Removing patches with no faces in them.

Removing zero-sized patch defaultFaces type empty at position 0
Removing zero-sized patch rotate_area_Mesh type wall at position 6
Removing zero-sized patch rotate_area_Mesh_slave type wall at position 7
Removing patches.
Writing repatched mesh to 3

End
    
```



2013.6.15

62/83

# 4-1. メッシュの作成(AMI)

## ⑨createBaffles 回転面の設定

```

sakuramaru@SAKURA-MARU: ~/Desktop/ver_2.2.x/small_fan_AMI$ createBaffles
-----*-----
      F ield      | OpenFOAM: The Open Source CFD Toolbox
      O peration  | Version: 2.2.x
      A nd        | Web: www.OpenFOAM.org
      M anipulation|
-----*-----
Build   : 2.2.x-b477bcabd47f
Exec    : createBaffles
Date    : May 11 2013
Time    : 18:32:32
Host    : "SAKURA-MARU"
PID     : 17937
Case    : /home/sakuramaru/Desktop/ver_2.2.x/small_fan_AMI
nProcs  : 1
sigFpe  : Enabling floating point exception trapping (FOAM_SIGFPE).
fileModificationChecking : Monitoring run-time modified files using timeStampMaster
allowSystemOperations : Disallowing user-supplied system call operations

// *****
Create time

Create mesh for time = 3

Reading baffle criteria from createBafflesDict

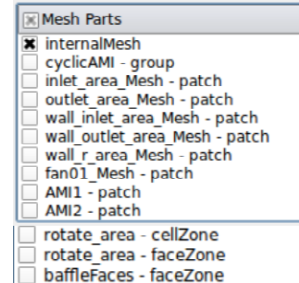
Not converting faces on non-coupled patches.

Reading geometric fields

Created zone baffleFaces at index 1 with 46594 faces
Patch 'AMI1' already exists. Only moving patch faces - type will remain the same
Patch 'AMI2' already exists. Only moving patch faces - type will remain the same
Converted 46594 faces into boundary faces in patches
2
(
AMI1
AMI2
)
    
```

```

--> FOAM Warning :
From function createBaffles
in file createBaffles.C at line 657
Setting field on boundary faces to zero.
You might have to edit these fields.
Writing mesh to 4
End
    
```

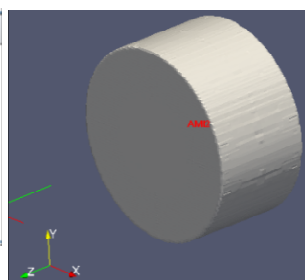
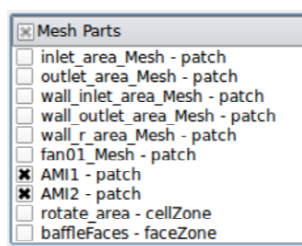
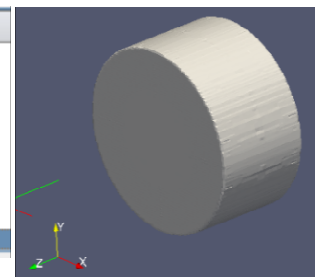
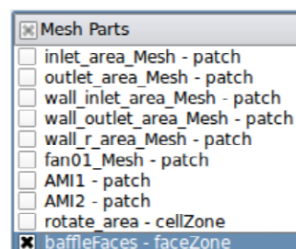
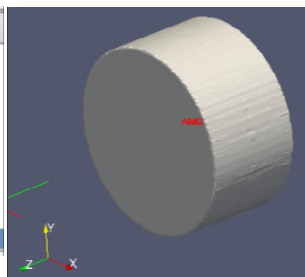
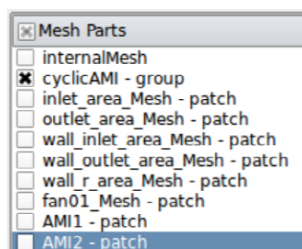


2013.6.15

63/83

# 4-1. メッシュの作成(AMI)

## ⑨createBaffles 回転面の設定



2013.6.15

64/83



## 4-1. メッシュの作成 (AMI)

## ⑩mergeOrSplitBaffles 回転面の分割

```
sakuramaru@SAKURA-MARU:~/Desktop/ver_2.2.x/small_fan_AMI$ mergeOrSplitBaffles -split
-----
Field          | OpenFOAM: The Open Source CFD Toolbox
Operation      | Version: 2.2.x
And            | Web: www.OpenFOAM.org
Manipulation   |
-----
Build : 2.2.x-b477bcabd47f
Exec : mergeOrSplitBaffles -split
Date : May 11 2013
Time : 19:35:56
Host : "SAKURA-MARU"
PID : 17995
Case : /home/sakuramaru/Desktop/ver_2.2.x/small_fan_AMI
nProcs : 1
sigFpe : Enabling floating point exception trapping (FOAM_SIGFPE).
fileModificationChecking : Monitoring run-time modified files using timeStampMaster
allowSystemOperations : Disallowing user-supplied system call operations

// *****
Create time

Create mesh for time = 4

Topologically splitting duplicate surfaces, i.e. duplicating points internal to duplicate surfaces.

Writing mesh to time 5
Writing 97342 duplicated points to pointSet "/home/sakuramaru/Desktop/ver_2.2.x/small_fan_AMI/5/polyMesh/sets/duplicatedPoints"

End
```

2013.6.15

65/83

## 4-1. メッシュの作成 (AMI)

## ⑪checkMesh チェックメッシュ

```
Checking geometry...
Overall domain bounding box (-0.284872 -0.284865 -0.235544) (0.284872 0.28489 0.534522)
Mesh (non-empty, non-wedge) directions (1 1 1)
Mesh (non-empty) directions (1 1 1)
Boundary openness (-3.48343e-16 3.85831e-16 -5.69087e-15) OK.
Max cell openness = 3.23629e-16 OK.
Max aspect ratio = 4.76655 OK.
Minimum face area = 1.79334e-08. Maximum face area = 0.000178993. Face area magnitudes OK.
Min volume = 5.42599e-12. Max volume = 1.33123e-06. Total volume = 0.0786396. Cell volumes OK.
Mesh non-orthogonality Max: 54.6084 average: 8.75087
Non-orthogonality check OK.
Face pyramids OK.
Max skewness = 3.73333 OK.
Coupled point location match (average 0) OK.
```

Mesh OK.

今回のAMI計算で利用したメッシュの品質

2013.6.15

66/83

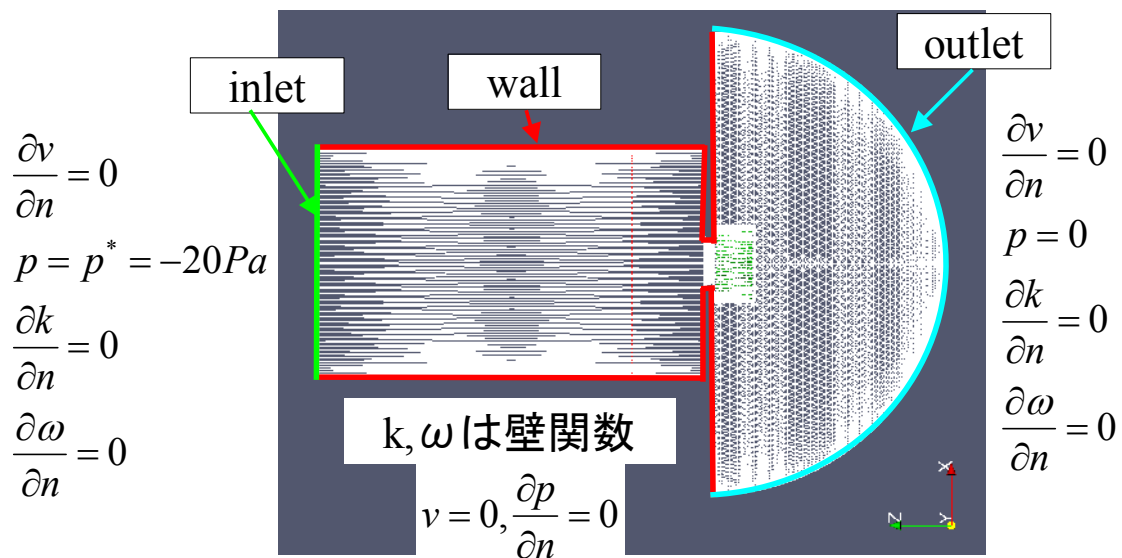
# 4-2. 計算の設定 AMI

2013.6.15

67/83

## 4-2. 計算の設定 (MRFと同じ)

### 境界条件



2013.6.15

68/83



## 4-2. 計算の設定

乱流モデル:kOmegaSST

回転速度:3000rpm(1回転は0.02秒)

dynamicMeshDictで設定

```
dynamicFvMesh    solidBodyMotionFvMesh;

motionSolverLibs ( "libfvMotionSolvers.so" );

solidBodyMotionFvMeshCoeffs
{
    cellZone      rotate_area;

    solidBodyMotionFunction    rotatingMotion;
    rotatingMotionCoeffs
    {
        CofG          (0 0 0);
        radialVelocity (0 0 -18000); // deg/s
    }
}
}
```

### MRF

```
MRF1
{
    type            MRFSource;
    active          true;
    selectionMode   cellZone;
    cellZone        rotate_area;

    MRFSourceCoeffs
    {
        origin      (0 0 0);
        axis        (0 0 -1);
        omega       314; //3000rpm
    }
}
```

MRFと回転数の設定単位が異なる

2013.6.15

69/83

## 4-2. 計算の設定

fvSolution

```
solvers
{
    pccorr
    {
        solver          GAMG;
        tolerance       1e-2;
        relTol          0;
        smoother        DICGaussSeidel;
        cacheAgglomeration no;
        nCellsInCoarsestLevel 10;
        agglomerator    faceAreaPair;
        mergeLevels     1;
        maxIter         50;
    }

    p
    {
        $pccorr;
        tolerance       1e-5;
        relTol          0.01;
    }

    pFinal
    {
        $p;
        tolerance       1e-6;
        relTol          0;
    }

    "(U|k|epsilon|omega)"
    {
        solver          smoothSolver;
        smoother        GaussSeidel;
        tolerance       1e-6;
        relTol          0.1;
    }

    "(U|k|epsilon|omega)Final"
    {
        solver          PBiCG;
        preconditioner  DILU;
        tolerance       1e-6;
        relTol          0;
    }
}

PIMPLE
{
    correctPhi          no;
    nOuterCorrectors   2;
    nCorrectors         1;
    nNonOrthogonalCorrectors 0;
}

relaxationFactors
{
    "(U|k|epsilon|omega).*"  1;
}

cache
{
    grad(U);
}
```

2013.6.15

70/83

## 4-2. 計算の設定

### fvSchemes

```
ddtSchemes
{
  default Euler;
}

gradSchemes
{
  default Gauss linear;
  grad(p) Gauss linear;
  grad(U) Gauss linear;
}

divSchemes
{
  default none;
  div(phi,U) bounded Gauss upwind;
  div(phi,k) bounded Gauss upwind;
  div(phi,epsilon) bounded Gauss upwind;
  div(phi,omega) bounded Gauss upwind;

  div((nuEff*dev(T(grad(U)))) Gauss linear;
}

laplacianSchemes
{
  default Gauss linear limited 0.5;
}

laplacianSchemes
{
  default Gauss linear limited 0.5;
}

interpolationSchemes
{
  default linear;
  interpolate(U) linear;
}

snGradSchemes
{
  default limited corrected 0.33;
}

fluxRequired
{
  default no;
  pcorr ;
  p ;
}
```

2013.6.15

71/83

## 4-3. 計算の結果

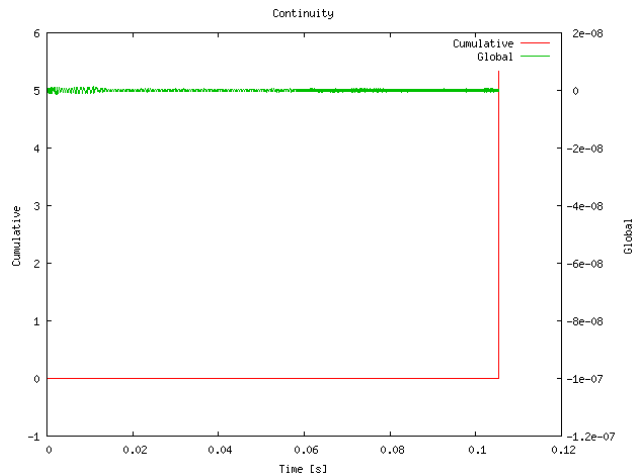
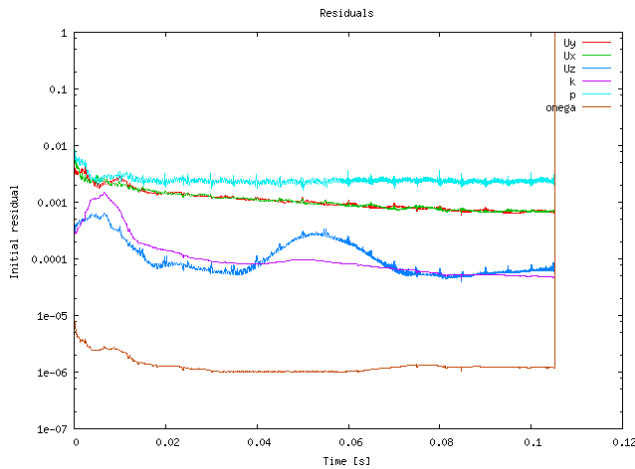
# AMI

2013.6.15

72/83

### 4-3. 計算の結果

・チュートリアルの方法でメッシュ作成すると、5回転しても計算途中で突然エラーする。



2013.6.15

73/83

### 4-3. 計算の結果

・突然のエラーはこんな感じ。

```
Courant Number mean: 0.00203907 max: 1.96486
deltaT = 1.81509e-05
Time = 0.105273
```

```
solidBodyMotionFunctions::rotatingMotion::transformation(): Time = 0.105273 transformation: ((0 0 0) (-0.67611 (0 0 0.7368)))
AMI: Creating addressing and weights between 59086 source faces and 59086 target faces
AMI: Patch source weights min/max/average = 0.896255, 1.28056, 1.00182
AMI: Patch target weights min/max/average = 0.828582, 1.28274, 1.00187
PIMPLE: iteration 1
smoothSolver: Solving for Ux, Initial residual = 0.000675985, Final residual = 4.25875e-05, No Iterations 1
smoothSolver: Solving for Uy, Initial residual = 0.000651236, Final residual = 4.10634e-05, No Iterations 1
smoothSolver: Solving for Uz, Initial residual = 5.98836e-05, Final residual = 1.69283e-06, No Iterations 1
GAMG: Solving for p, Initial residual = 0.00236927, Final residual = 2.16976e-05, No Iterations 4
time step continuity errors : sum local = 1.34285e-09, global = 4.10911e-10, cumulative = 2.60918e-07
PIMPLE: iteration 2
DILUPBiCG: Solving for Ux, Initial residual = 1.15386e-05, Final residual = 1.14971e-07, No Iterations 2
DILUPBiCG: Solving for Uy, Initial residual = 1.20183e-05, Final residual = 1.52632e+10, No Iterations 1001
DILUPBiCG: Solving for Uz, Initial residual = 9.83078e-07, Final residual = 9.83078e-07, No Iterations 0
GAMG: Solving for p, Initial residual = 1, Final residual = 6.25543e-07, No Iterations 16
time step continuity errors : sum local = 22.8904, global = 5.33073, cumulative = 5.33073
DILUPBiCG: Solving for omega, Initial residual = 1, Final residual = 6.90463, No Iterations 1001
bounding omega, min: -3.80568e+22 max: 3.30341e+22 average: 1.53801e+19
DILUPBiCG: Solving for k, Initial residual = 0.5878, Final residual = 5.4565e-07, No Iterations 2
bounding k, min: -0.102468 max: 31.0401 average: 3.86844
ExecutionTime = 79675.1 s ClockTime = 80455 s
```

```
Courant Number mean: 8.53346e+07 max: 5.00768e+14
deltaT = 7.24922e-20
--> FOAM Warning :
From function Time::operator++()
in file db/Time/Time.C at line 1029
Increased the timePrecision from 6 to 7 to distinguish between timeNames at time 0.105273
Time = 0.1052733
Duration of pickling 0.731229782104 too long. Extending frequency from 1.0 to 36.5614891052
```

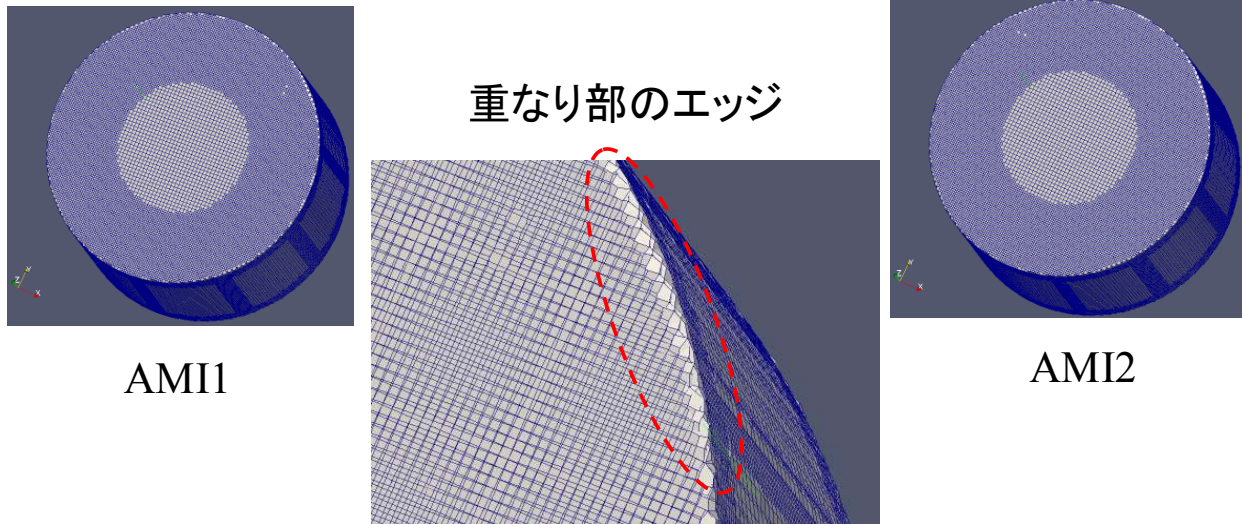
**クーラン条件が爆発**

2013.6.15

74/83

### 4-3. 計算の結果

- ・計算途中で突然エラーする問題は、回転部分のメッシュがやはり問題の可能性はある？。

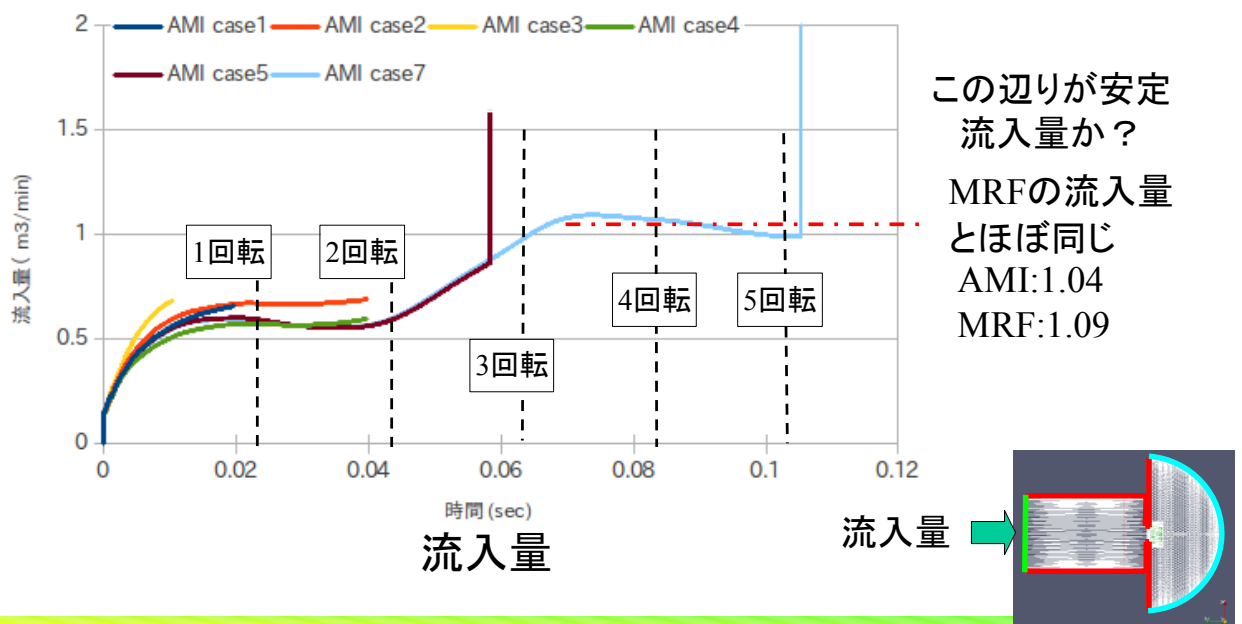


2013.6.15

75/83

### 4-3. 計算の結果

- ・流量が0.02秒で一旦平衡になり0.04秒後に増え始める理由は何か？
- ・流入量を見積もるには何回転程度必要か？

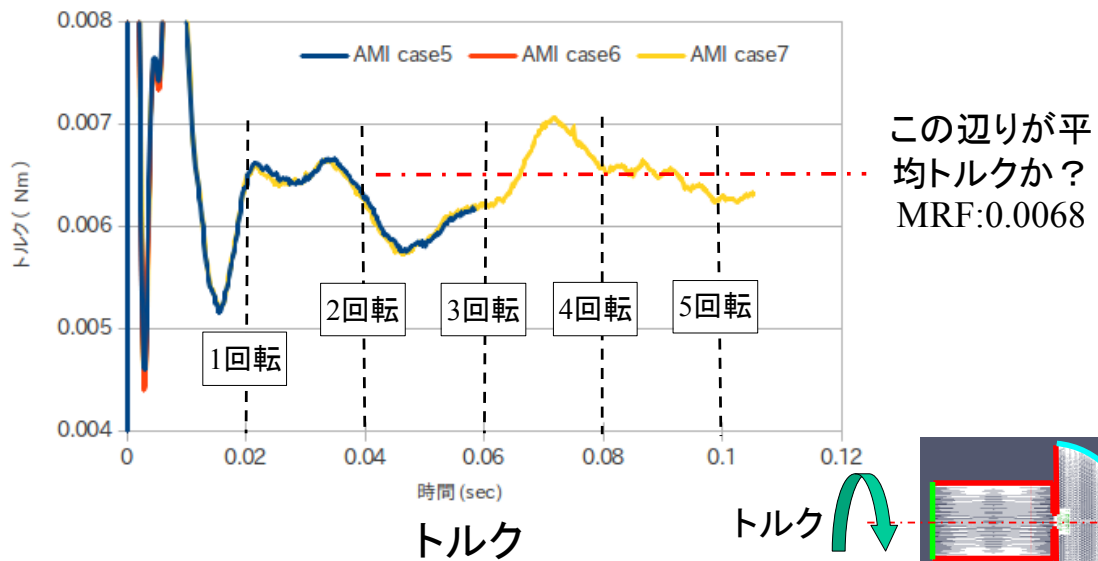


2013.6.15

76/83

### 4-3. 計算の結果

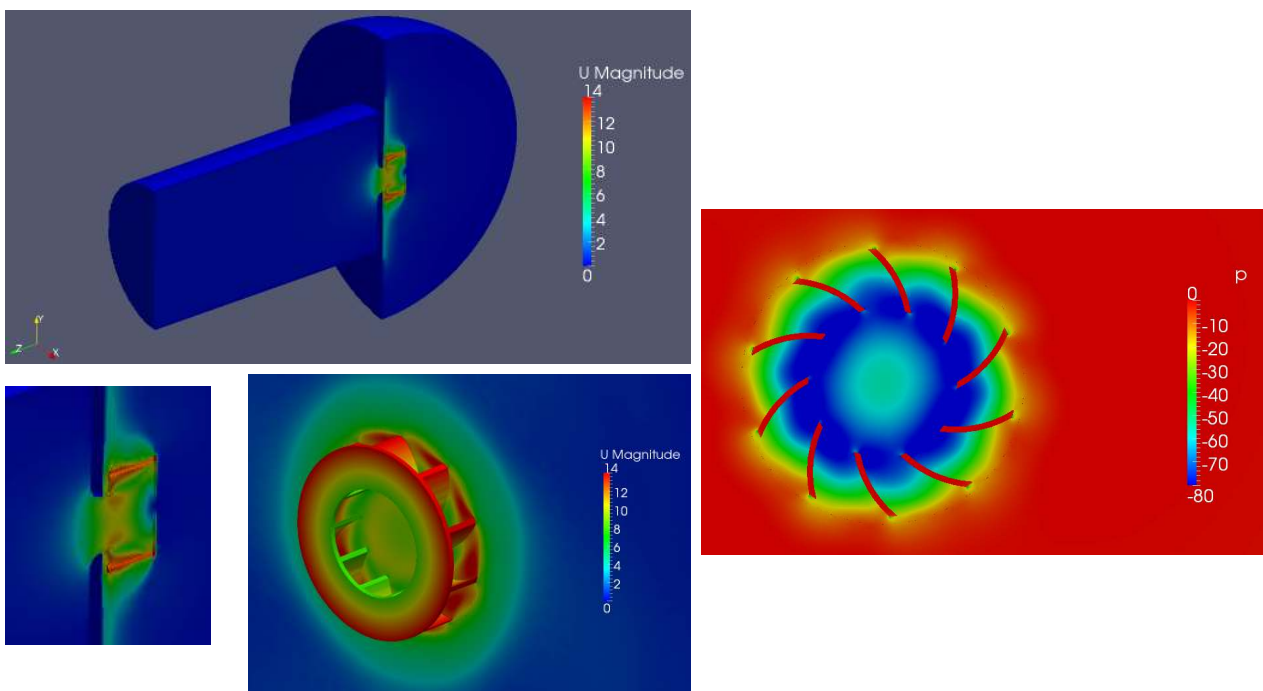
- ・流入量以上にトルクの変動は大きい。
- ・何回転位させたら安定するか？



2013.6.15

77/83

### 4-3. 計算の結果



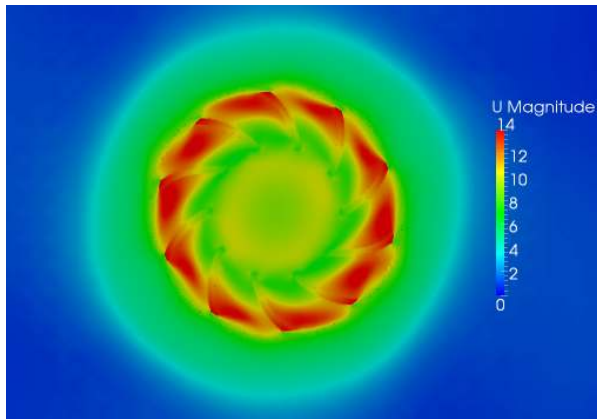
2013.6.15

78/83

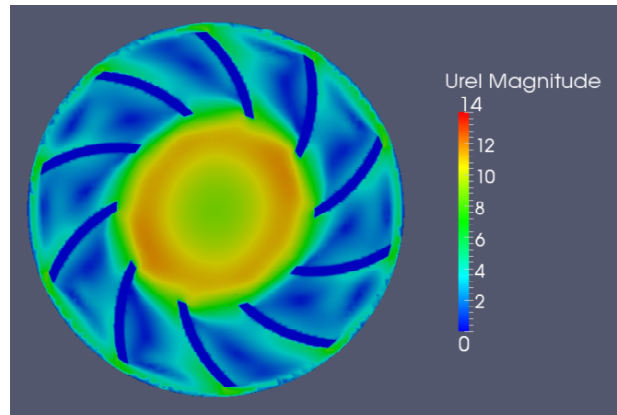


### 4-3. 計算の結果

回転座標系での速度表示もMRFと同じようにUrelで出来る。



全体座標系

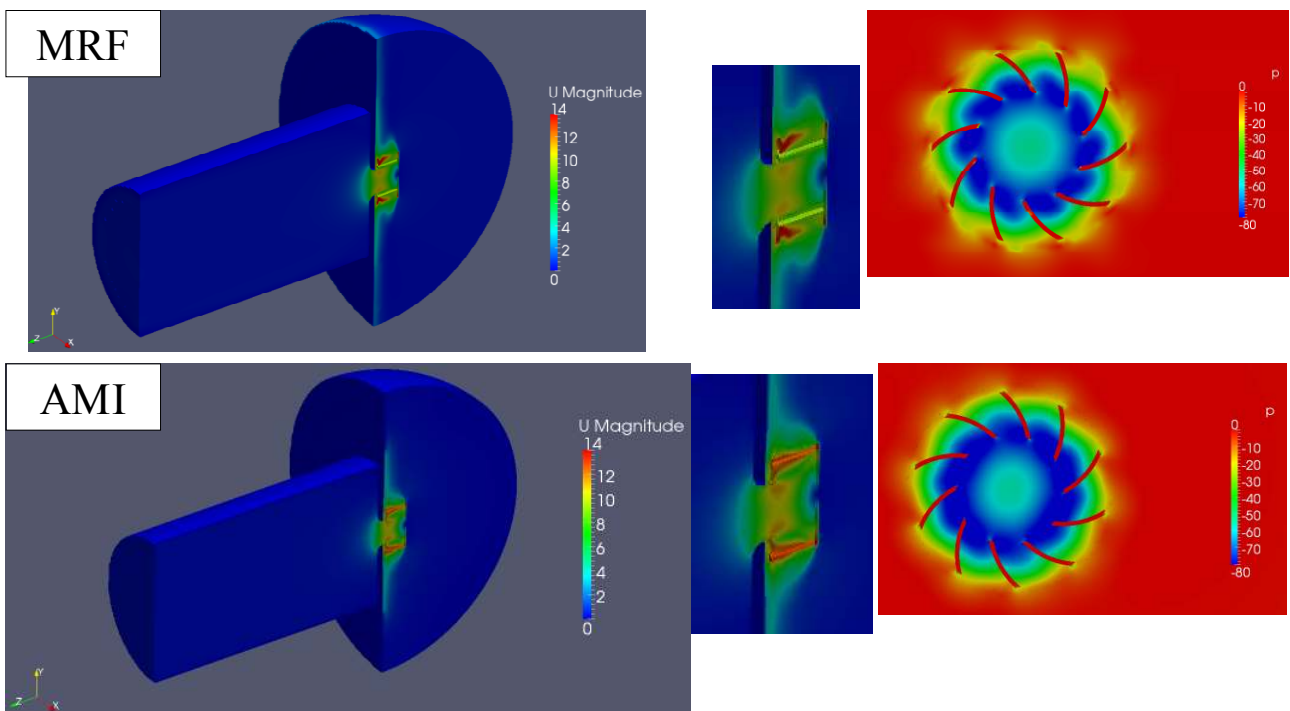


回転座標系

2013.6.15

79/83

### 4-3. 計算の結果 AMIとMRFの差は？

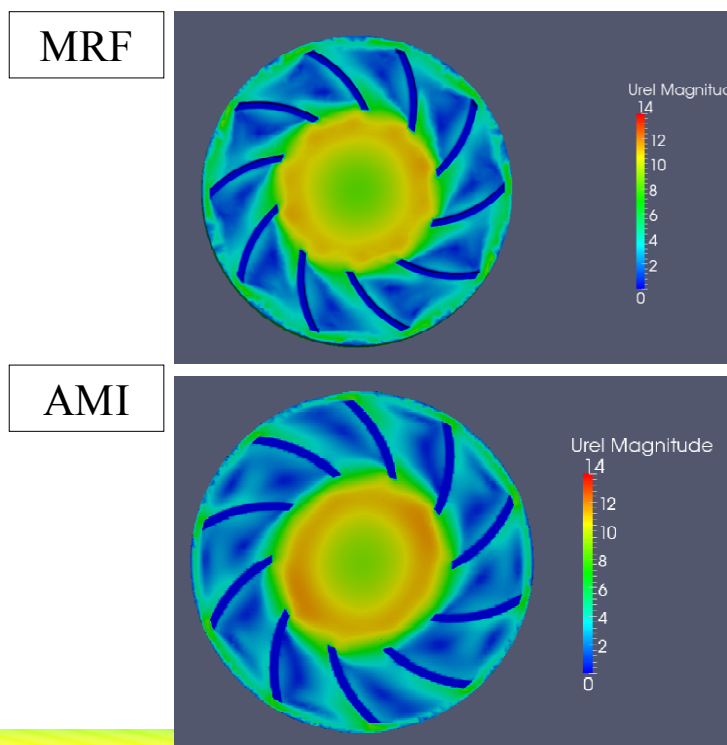


2013.6.15

80/83



### 4-3. 計算の結果 AMIとMRFの差は？



2013.6.15

81/83

## 5. まとめ, その他

2013.6.15

82/83

## 5. まとめ, その他

- ・フリーソフトを乗り継ぐ事で, モデリングから計算まで勉強レベルの流体解析は十分に出来る。
- ・MRF領域の大きさが多少変化しても, PQ特性にはあまり影響を及ぼさない。
- ・2.2から導入されたbounded Gauss upwindは収束が良い。(この事例検討では)
- ・AMIはメッシュ品質の影響を受けやすい。突然計算が発散する。
- ・AMIの計算時間はMRFに比べてかなり時間がかかる。(実用的にはMRFの利用が良い)