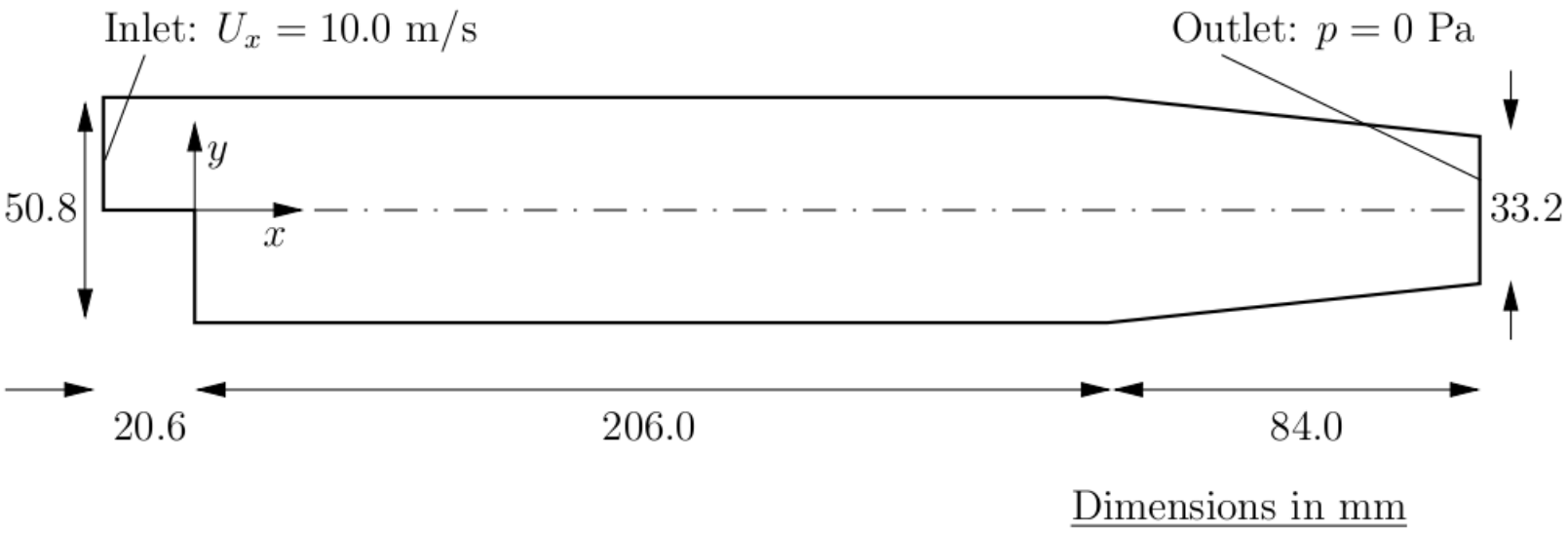
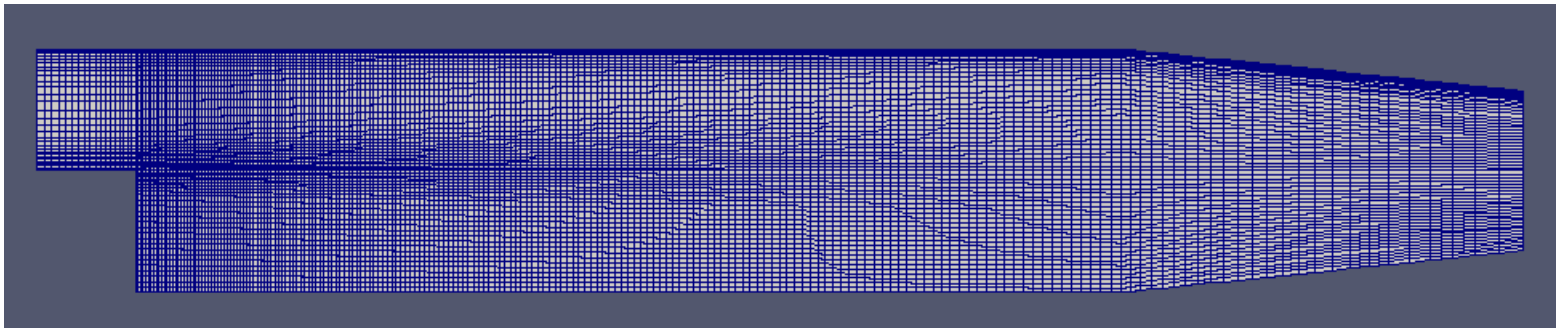
（1）バックステップ流れ

1．モデル，メッシュ形状
2．計算に利用したRANS系の乱流モデル
3．境界条件
4．数値スキーム　fvSchemes
5．解法とアルゴリズム制御　fvSolution
6．計算設定
7．y+の疑問
8．計算結果の比較（RANS）
9．LESでの計算
10．DEXCS 2010を利用したバックステップ流れの操作
11．参考資料

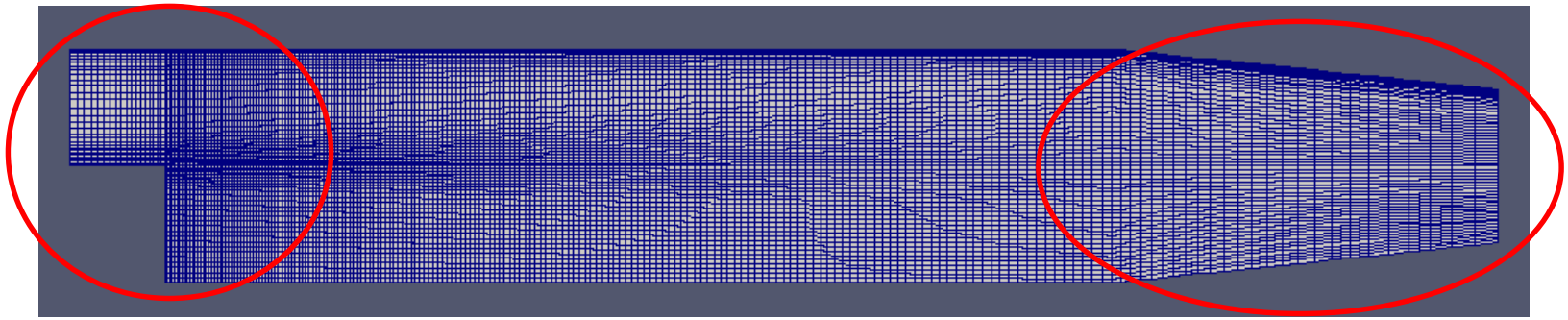
（2）質疑・応答

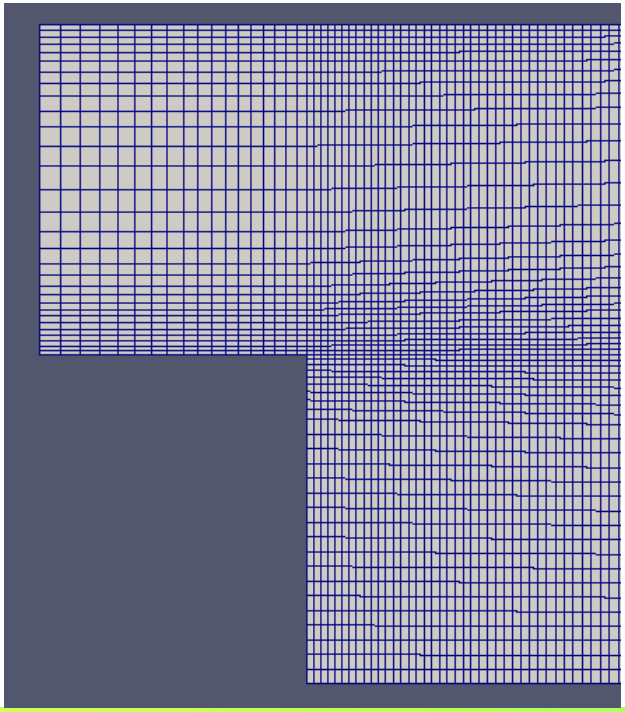
1．モデル，メッシュ形状（tutorials/incompressible/simpleFoam/pitzDaily）

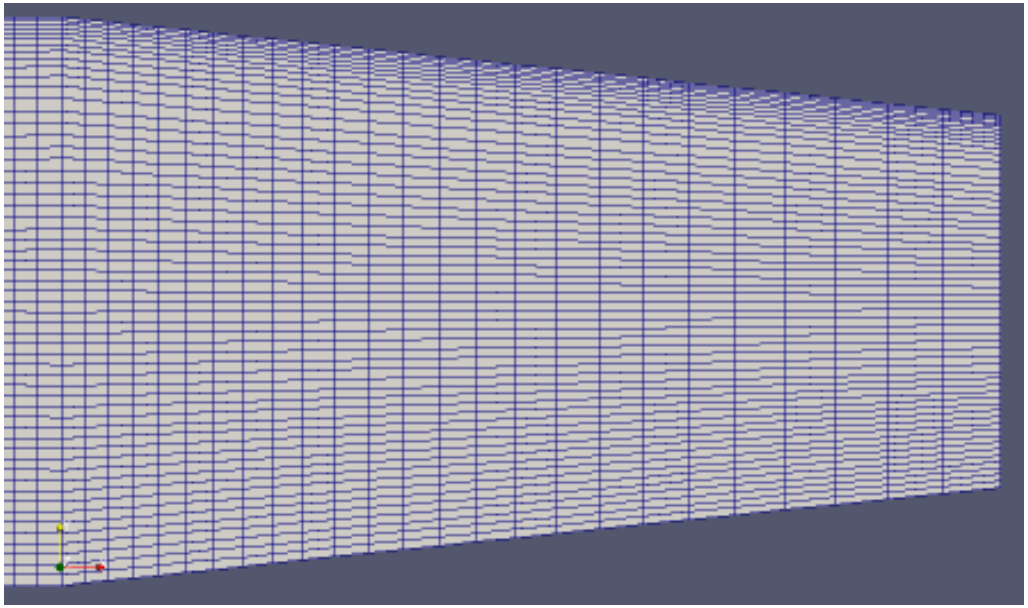




1．モデル，メッシュ形状







2. 計算に利用したRANS系の乱流モデル

次の3モデル（非圧縮）で計算を実施。
この3つを選択した理由は？

RAS turbulence models for incompressible fluids — incompressibleRASModels

laminar	Dummy turbulence model for laminar flow
kEpsilon	Standard high-Re $k-\varepsilon$ model
kOmega	Standard high-Re $k-\omega$ model
kOmegaSST	$k-\omega$-SST model
RNGkEpsilon	RNG $k-\varepsilon$ model
NonlinearKEShih	Non-linear Shih $k-\varepsilon$ model
LienCubicKE	Lien cubic $k-\varepsilon$ model
qZeta	$q-\zeta$ model
LaunderSharmaKE	Launder-Sharma low-Re $k-\varepsilon$ model
LamBremhorstKE	Lam-Bremhorst low-Re $k-\varepsilon$ model
LienCubicKELowRe	Lien cubic low-Re $k-\varepsilon$ model
LienLeschzinerLowRe	Lien-Leschziner low-Re $k-\varepsilon$ model
LRR	Launder-Reece-Rodi RSTM
LaunderGibsonRSTM	Launder-Gibson RSTM with wall-reflec
realizableKE	Realizable $k-\varepsilon$ model
SpalartAllmaras	Spalart-Allmaras 1-eqn mixing-length model

一番有名，チュートリアルでの乱流モデル

複雑な形状で発散し難い

k-εより優れている？

２．RANS系の乱流モデル

次の資料が色々と参考になるかと。。。

①http://www.cfd-online.com/Wiki/Turbulence_modeling

②CFD of Air Flow in Hydro Power Generators appendix A

A.1.2　kEpsilon

The kEpsilon model is the standard high-Re $k-\varepsilon$ model with wall functions [2]. It is implemented in OpenFOAM as

$$\frac{\partial k}{\partial t} + \nabla \cdot (Uk) - (\nabla \cdot U)k - \nabla \cdot D_{k,eff} \nabla k = G - \varepsilon$$

$$\frac{\partial \varepsilon}{\partial t} + \nabla \cdot (U\varepsilon) - (\nabla \cdot U)\varepsilon - \nabla \cdot D_{\varepsilon,eff} \nabla \varepsilon = C_{\varepsilon 1} \frac{\varepsilon}{k} G - C_{\varepsilon 2} \frac{\varepsilon^2}{k}$$

G	$D_{k,eff}$	$D_{\varepsilon,eff}$	α_ε	ν_t	$C_{\varepsilon 1}$	$C_{\varepsilon 2}$	C_μ
$\nu_t 2\lvert S_{ij}\rvert^2$	$\nu + \nu_t$	$\nu + \alpha_\varepsilon \nu_t$	0.76923	$C_\mu \frac{k^2}{\varepsilon}$	1.44	1.92	0.09

②の資料より

3. 境界条件

表に示すように山ほどの設定がある。適切に設定するガイドラインや使い方の説明が欲しい。。。

1.7.1ユーザーガイドにの5.2.4に簡単な説明がある程度！
後はソースを読めという事か？

みなさんどうしてますか？

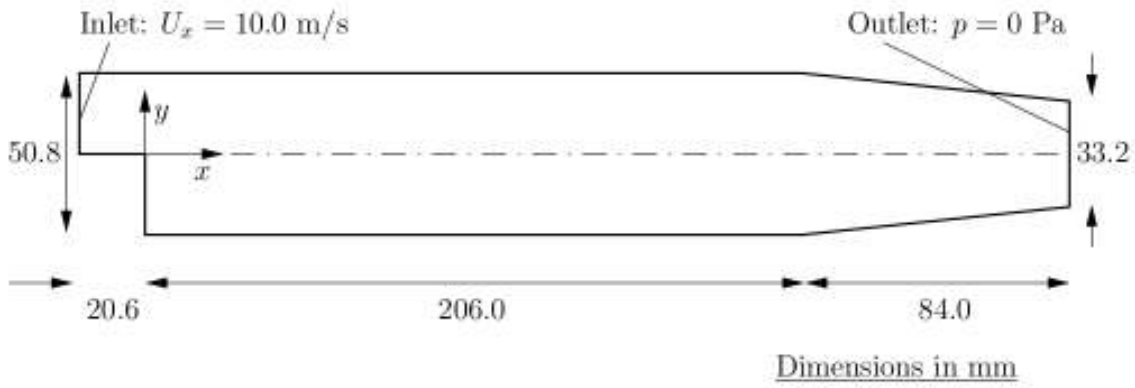
速度境界設定：54

圧力境界設定：57

K，ε，ω境界設定：57

	U	p	k，ε，ω
1	SRFFreestreamVelocity	advective	advective
2	SRFVelocity	atmBoundaryLayerInletEpsilon	atmBoundaryLayerInletEpsilon
3	activeBaffleVelocity	buoyantPressure	buoyantPressure
4	advective	calculated	calculated
5	atmBoundaryLayerInletVelocity	cyclic	cyclic
6	calculated	directMapped	directMapped
7	cyclic	directionMixed	directionMixed
8	cylindricalInletVelocity	empty	empty
9	directMapped	epsilonWallFunction	epsilonWallFunction
10	directMappedVelocityFlux	fan	fan
11	directionMixed	fixedFluxPressure	fixedFluxPressure
12	empty	fixedGradient	fixedGradient
13	fixedGradient	fixedInternalValue	fixedInternalValue
14	fixedInternalValue	fixedPressureCompressibleDensity	fixedPressureCompressibleDensity
15	fixedNormalSlip	fixedValue	fixedValue
16	fixedShearStress	freestream	freestream
17	fixedValue	freestreamPressure	freestreamPressure
18	flowRateInletVelocity	inletOutlet	inletOutlet
19	fluxCorrectedVelocity	inletOutletTotalTemperature	inletOutletTotalTemperature
20	freestream	kappatJayatillekeWallFunction	kappatJayatillekeWallFunction
21	inletOutlet	kqRWallFunction	kqRWallFunction
22	kqRWallFunction	mixed	mixed
23	mixed	nutLowReWallFunction	nutLowReWallFunction
24	movingWallVelocity	nutRoughWallFunction	nutRoughWallFunction
25	oscillatingFixedValue	nutSpalartAllmarasStandardRoughWallFunction	nutSpalartAllmarasStandardRoughWallFunction
26	outletInlet	nutSpalartAllmarasStandardWallFunction	nutSpalartAllmarasStandardWallFunction
27	outletMappedUniformInlet	nutSpalartAllmarasWallFunction	nutSpalartAllmarasWallFunction
28	partialSlip	nutWallFunction	nutWallFunction
29	pressureDirectedInletOutletVelocity	nutkWallFunction	nutkWallFunction
30	pressureDirectedInletVelocity	omegaWallFunction	omegaWallFunction
31	pressureInletOutletVelocity	oscillatingFixedValue	oscillatingFixedValue
32	pressureInletUniformVelocity	outletInlet	outletInlet
33	pressureInletVelocity	outletMappedUniformInlet	outletMappedUniformInlet
34	pressureNormalInletOutletVelocity	partialSlip	partialSlip
35	processor	processor	processor
36	rotatingPressureInletOutletVelocity	rotatingTotalPressure	rotatingTotalPressure
37	rotatingWallVelocity	sliced	sliced
38	sliced	slip	slip
39	slip	symmetryPlane	symmetryPlane
40	supersonicFreestream	syringePressure	syringePressure
41	surfaceNormalFixedValue	timeVaryingMappedFixedValue	timeVaryingMappedFixedValue
42	swirlFlowRateInletVelocity	timeVaryingMappedTotalPressure	timeVaryingMappedTotalPressure
43	symmetryPlane	timeVaryingTotalPressure	timeVaryingTotalPressure
44	timeVaryingFlowRateInletVelocity	timeVaryingUniformFixedValue	timeVaryingUniformFixedValue
45	timeVaryingMappedFixedValue	timeVaryingUniformInletOutlet	timeVaryingUniformInletOutlet
46	timeVaryingMappedPressureDirectedInletVelocity	totalPressure	totalPressure
47	timeVaryingUniformFixedValue	totalTemperature	totalTemperature
48	timeVaryingUniformInletOutlet	turbulentHeatFluxTemperature	turbulentHeatFluxTemperature
49	translatingWallVelocity	turbulentInlet	turbulentInlet
50	turbulentInlet	turbulentIntensityKineticEnergyInlet	turbulentIntensityKineticEnergyInlet
51	uniformFixedValue	turbulentMixingLengthDissipationRateInlet	turbulentMixingLengthDissipationRateInlet
52	waveTransmissive	turbulentMixingLengthFrequencyInlet	turbulentMixingLengthFrequencyInlet
53	wedge	uniformDensityHydrostaticPressure	uniformDensityHydrostaticPressure
54	zeroGradient	uniformFixedValue	uniformFixedValue
55		waveTransmissive	waveTransmissive
56		wedge	wedge
57		zeroGradient	zeroGradient

３．境界条件



インレット側　　速度　（10，0，0）　圧力　勾配＝0
アウトレット側　速度　勾配＝0　　圧力　大気開放
k，εについては，チュートリアルに設定方法が書いてある。

that described in section 2.1.8.1 of the User Guide. We assume that the inlet turbulence is isotropic and estimate the fluctuations to be 5% of **U** at the inlet. We have

$$U'_x = U'_y = U'_z = \frac{5}{100}10 = 0.5 \text{ m/s} \tag{3.6}$$

and

$$k = \frac{3}{2}(0.5)^2 = 0.375 \text{ m}^2/\text{s}^2 \tag{3.7}$$

If we estimate the turbulent length scale l to be 10% of the width of the inlet then

$$\varepsilon = \frac{C_\mu^{0.75} k^{1.5}}{l} = \frac{0.09^{0.75} 0.375^{1.5}}{0.1 \times 25.4 \times 10^{-3}} = 14.855 \text{ m}^2/\text{s}^3 \tag{3.8}$$

3. 境界条件

http://www.cfd-online.com/Wiki/Turbulence_free-stream_boundary_conditions

①I（turbulence intensity ）今回は5%で設定

High-turbulence case:

High-speed flow inside complex geometries like heat-exchangers and flow inside rotating machinery (turbines and compressors). Typically the turbulence intensity is between 5% and 20%

Medium-turbulence case:

Flow in not-so-complex devices like large pipes, ventilation flows etc. or low speed flows (low Reynolds number). Typically the turbulence intensity is between 1% and 5%

Low-turbulence case:

Flow originating from a fluid that stands still, like external flow across cars, submarines and aircrafts. Very high-quality wind-tunnels can also reach really low turbulence levels. Typically the turbulence intensity is very low, well below 1%.

3. 境界条件

http://www.cfd-online.com/Wiki/Turbulence_free-stream_boundary_conditions

②k

$$k = \frac{3}{2}\,(U\,I)^2$$

Where U is the mean flow velocity and I is the turbulence intensity.

③ε

$$\epsilon = C_\mu^{\frac{3}{4}}\,\frac{k^{\frac{3}{2}}}{l}$$

lはTurbulence length scale

④ω

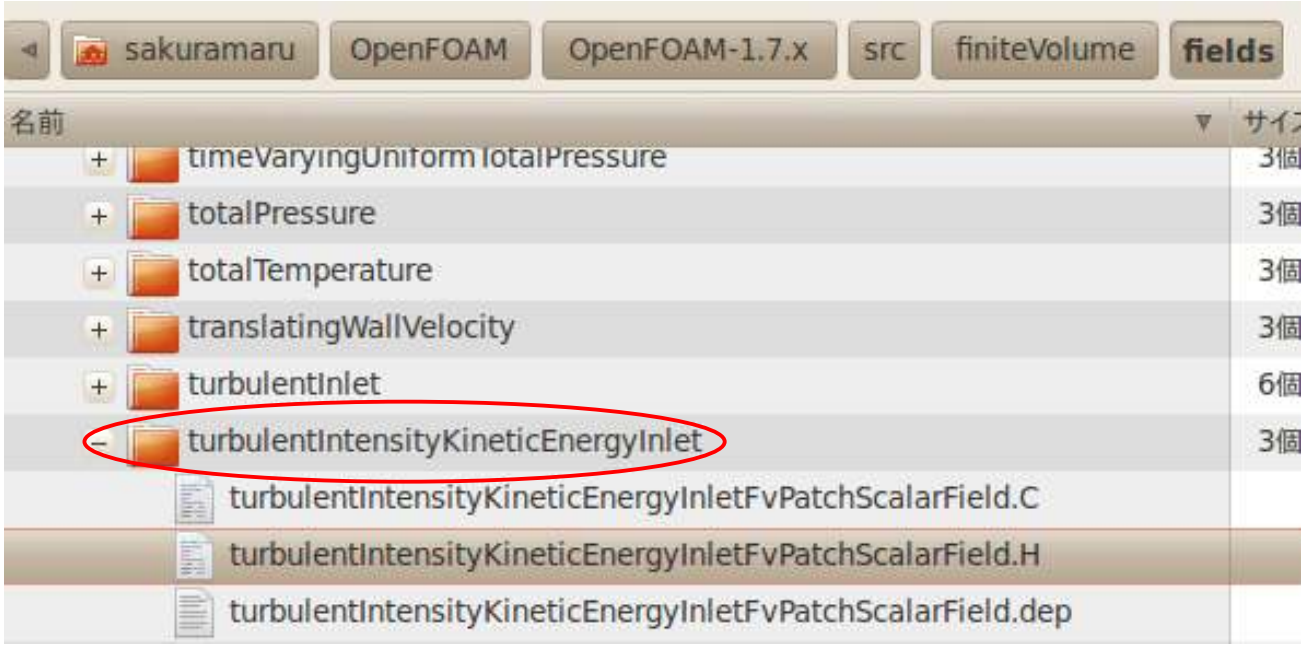
$$\omega = C_\mu^{-\frac{1}{4}}\,\frac{\sqrt{k}}{l}$$

３．境界条件

①k : turbulentIntensityKineticEnergyInlet

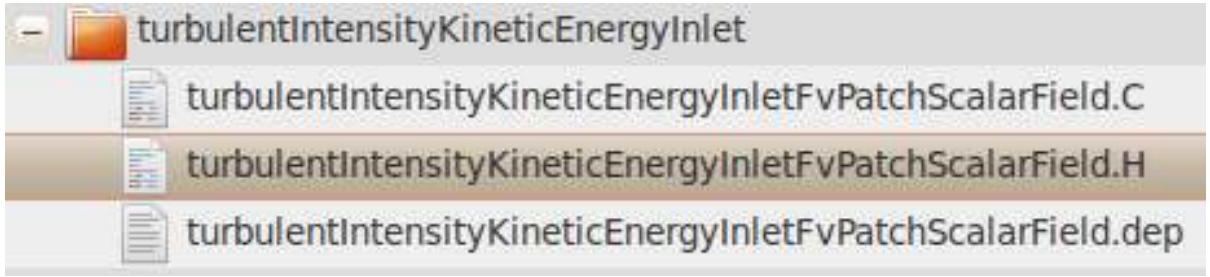
unixのコマンドで探す。find -name turbulentIntensityKineticEnergyInlet

./OpenFOAM-1.7.x/src/finiteVolume/fields/fvPatchFields/derived/turbulentIntensityKineticEnergyInlet



３．境界条件

①k : turbulentIntensityKineticEnergyInlet



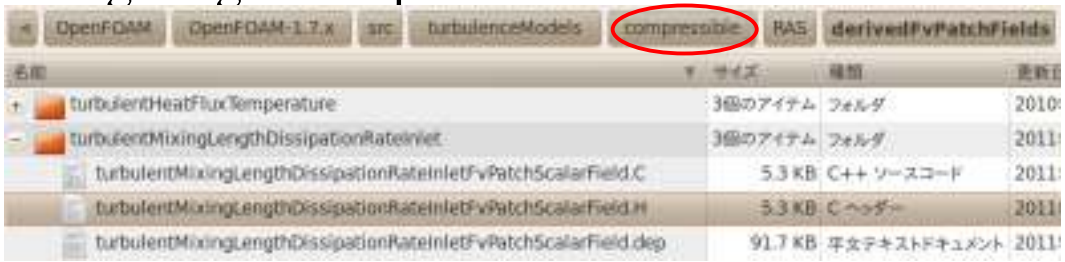

```
Description
    Calculate turbulent kinetic energy from the intensity provided as a
    fraction of the mean velocity

    Example of the boundary condition specification:
    @verbatim
     inlet
    {
         type        turbulentIntensityKineticEnergyInlet;
         intensity   0.05;            // 5% turbulence
         value       uniform 1;       // placeholder
    }
    @endverbatim
```

３．境界条件

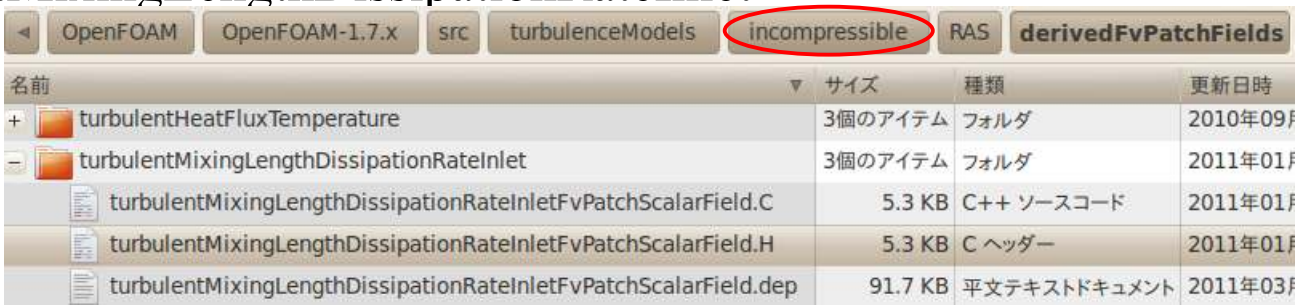
②epsilon : turbulentMixingLengthDissipationRateInlet

unixのコマンドで探す。find -name turbulentMixingLengthDissipationRateInlet

./OpenFOAM-1.7.x/src/turbulenceModels/compressible/RAS/derivedFvPatchFields/turbulentMixingLengthDissipationRateInlet



./OpenFOAM-1.7.x/src/turbulenceModels/incompressible/RAS/derivedFvPatchFields/turbulentMixingLengthDissipationRateInlet



３．境界条件

②epsilon : turbulentMixingLengthDissipationRateInlet

unixのコマンドで探す。find -name turbulentMixingLengthDissipationRateInlet

./OpenFOAM-1.7.x/src/turbulenceModels/compressible/RAS/derivedFvPatchFields/turbulentMixingLengthDissipationRateInlet

```
Description
    Calculate epsilon via the mixing length [m]

    Example of the boundary condition specification:
    @verbatim
        inlet
       {
            type            compressible::turbulentMixingLengthDissipationRateInlet;
            mixingLength    0.005;          // 5 mm
            value           uniform 200;    // placeholder
       }
    @endverbatim
```

./OpenFOAM-1.7.x/src/turbulenceModels/incompressible/RAS/derivedFvPatchFields/turbulentMixingLengthDissipationRateInlet

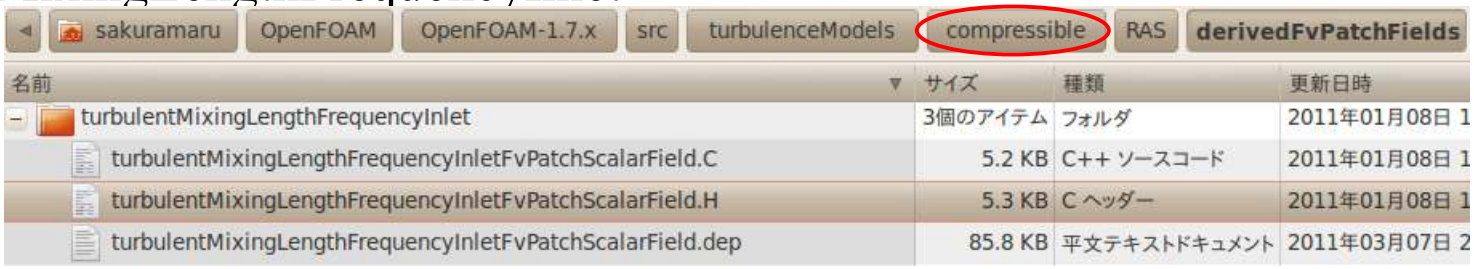
```
Description
    Calculate epsilon via the mixing length [m]

    Example of the boundary condition specification:
    @verbatim
        inlet
       {
            type            turbulentMixingLengthDissipationRateInlet;
            mixingLength    0.005;          // 5 mm
            value           uniform 200;    // placeholder
       }
    @endverbatim
```

３．境界条件

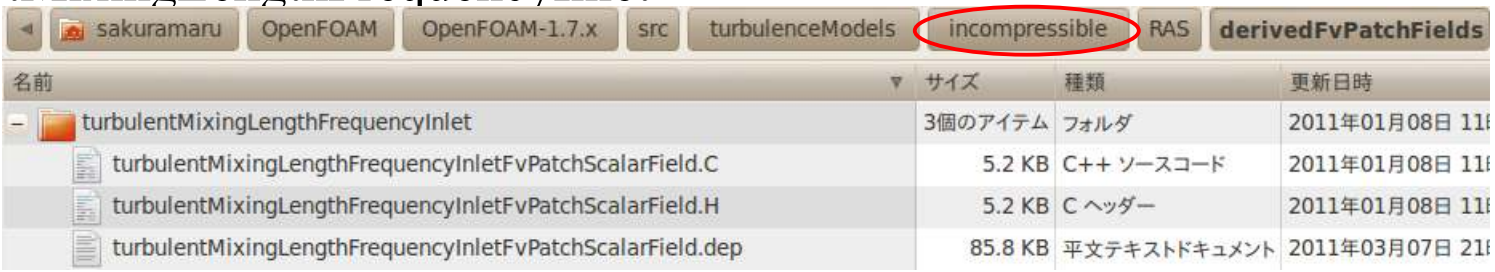
② omega : turbulentMixingLengthFrequencyInlet

unixのコマンドで探す。find -name turbulentMixingLengthFrequencyInlet

./OpenFOAM-1.7.x/src/turbulenceModels/compressible/RAS/derivedFvPatchFields/turbulentMixingLengthFrequencyInlet



./OpenFOAM-1.7.x/src/turbulenceModels/incompressible/RAS/derivedFvPatchFields/turbulentMixingLengthFrequencyInlet



３．境界条件

② omega : turbulentMixingLengthFrequencyInlet

unixのコマンドで探す。find -name turbulentMixingLengthFrequencyInlet

./OpenFOAM-1.7.x/src/turbulenceModels/compressible/RAS/derivedFvPatchFields/turbulentMixingLengthFrequencyInlet

```
Description
    Calculate omega via the mixing length

    Example of the boundary condition specification:
    @verbatim
        inlet
       {
            type            compressible::turbulentMixingLengthFrequencyInlet;
            mixingLength    0.005;          // 5 mm
            k               k;              // turbulent k field
            value           uniform 5;      // initial value
       }
    @endverbatim
```

./OpenFOAM-1.7.x/src/turbulenceModels/incompressible/RAS/derivedFvPatchFields/turbulentMixingLengthFrequencyInlet

```
Description
    Calculate omega via the mixing length

    Example of the boundary condition specification:
    @verbatim
        inlet
       {
            type            turbulentMixingLengthFrequencyInlet;
            mixingLength    0.005;          // 5 mm
            k               k;              // turbulent k field
            value           uniform 5;      // initial value
       }
    @endverbatim
```

３．境界条件

② omega : turbulentMixingLengthFrequencyInlet

unixのコマンドで探す。find -name turbulentMixingLengthFrequencyInlet

./OpenFOAM-1.7.x/src/turbulenceModels/compressible/RAS/derivedFvPatchFields/turbulentMixingLengthFrequencyInlet

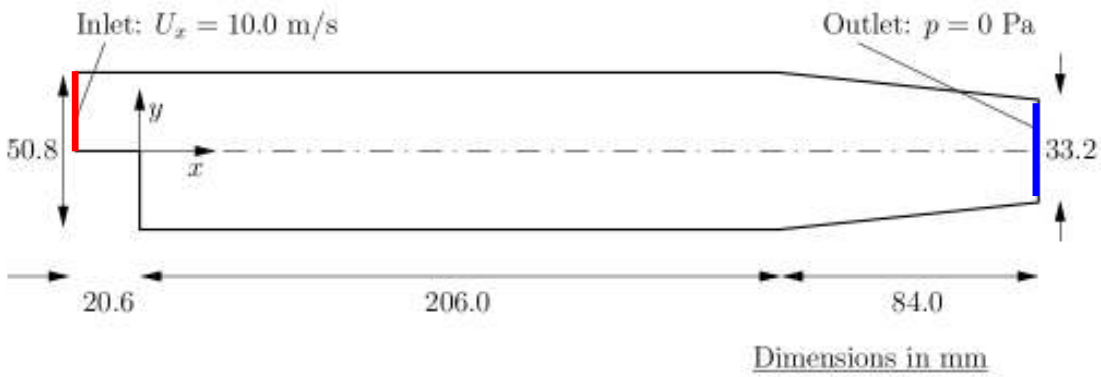
```
Description
    Calculate omega via the mixing length

    Example of the boundary condition specification:
    @verbatim
        inlet
       {
            type            compressible::turbulentMixingLengthFrequencyInlet;
            mixingLength    0.005;          // 5 mm
            k               k;              // turbulent k field
            value           uniform 5;      // initial value
       }
    @endverbatim
```

./OpenFOAM-1.7.x/src/turbulenceModels/incompressible/RAS/derivedFvPatchFields/turbulentMixingLengthFrequencyInlet

```
Description
    Calculate omega via the mixing length

    Example of the boundary condition specification:
    @verbatim
        inlet
       {
            type            turbulentMixingLengthFrequencyInlet;
            mixingLength    0.005;          // 5 mm
            k               k;              // turbulent k field
            value           uniform 5;      // initial value
       }
    @endverbatim
```

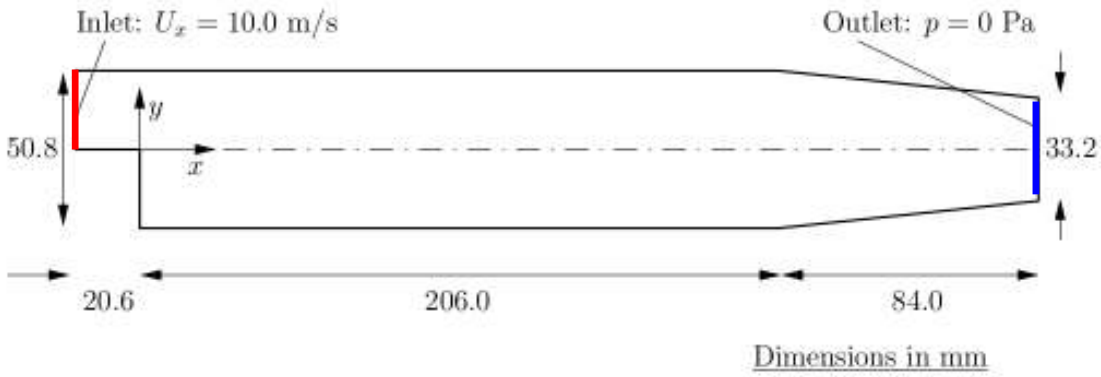
３. 境界条件：　圧力　ｐ

```
dimensions      [0 2 -2 0 0 0 0];

internalField   uniform 0;

boundaryField
{
    inlet
    {
        type            zeroGradient;
    }

    outlet
    {
        type            fixedValue;
        value           uniform 0;
    }

    upperWall
    {
        type            zeroGradient;
    }

    lowerWall
    {
        type            zeroGradient;
    }

    frontAndBack
    {
        type            empty;
    }
}
```



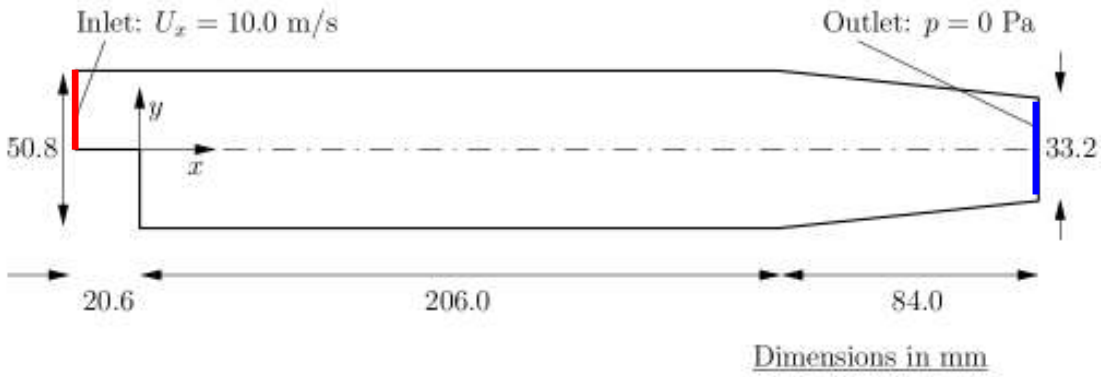
３. 境界条件：　速度　U

```
dimensions      [0 1 -1 0 0 0 0];

internalField   uniform (0 0 0);

boundaryField
{
    inlet
    {
        type          fixedValue;
        value         uniform (10 0 0);
    }

    outlet
    {
        type          zeroGradient;
    }

    upperWall
    {
        type          fixedValue;
        value         uniform (0 0 0);
    }

    lowerWall
    {
        type          fixedValue;
        value         uniform (0 0 0);
    }

    frontAndBack
    {
        type          empty;
    }
}
```



３．境界条件：　乱流エネルギー（その１）　k

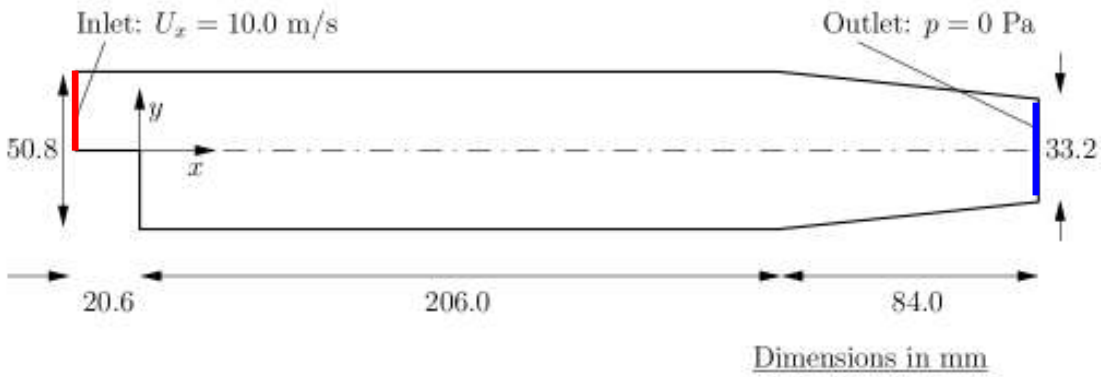
```
dimensions      [0 2 -2 0 0 0 0];

internalField   uniform 0.375;

boundaryField
{
    inlet
    {
        type            fixedValue;
        value           uniform 0.375;
    }
    outlet
    {
        type            zeroGradient;
    }
    upperWall
    {
        type            kqRWallFunction;
        value           uniform 0.375;
    }
    lowerWall
    {
        type            kqRWallFunction;
        value           uniform 0.375;
    }
    frontAndBack
    {
        type            empty;
    }
}
```



３．境界条件：　乱流エネルギー（その２）　k

```
dimensions      [0 2 -2 0 0 0 0];

internalField   uniform 0.375;

boundaryField
{
    inlet
    {
        type turbulentIntensityKineticEnergyInlet;
        intensity 0.05;
        value uniform 1;
    }
    outlet
    {
        type inletOutlet;
        inletValue uniform 0.375;
        value uniform 0.375;
    }
    upperWall
    {
        type        kqRWallFunction;
        value       uniform 0.375;
    }
    lowerWall
    {
        type        kqRWallFunction;
        value       uniform 0.375;
    }
    frontAndBack
    {
        type        empty;
    }
}
```

inletOutlet
ドメインに流入するときディリクレ条件
流出するときノイマン条件



３．境界条件：　散逸率（その１）　ε

```
dimensions      [0 2 -3 0 0 0 0];

internalField   uniform 14.855;

boundaryField
{
inlet
    {
        type          fixedValue;
        value          uniform 14.855;
    }
outlet
    {
        type          zeroGradient;
    }
upperWall
    {
        type          epsilonWallFunction;
        value          uniform 14.855;
    }
lowerWall
    {
        type          epsilonWallFunction;
        value          uniform 14.855;
    }
frontAndBack
    {
        type          empty;
    }
}
```

３．境界条件：　散逸率（その２）　ε

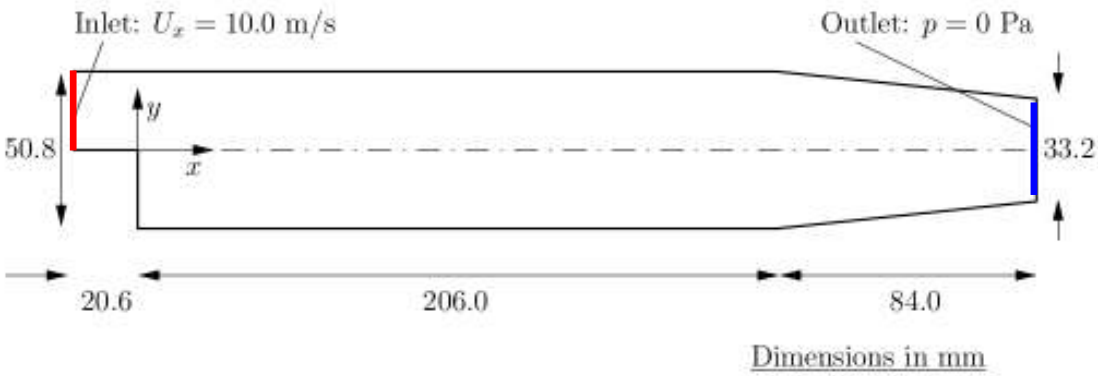
```
dimensions      [0 2 -3 0 0 0 0];

internalField   uniform 14.855;

boundaryField
{
    inlet
    {
        type turbulentMixingLengthDissipationRateInlet;
        mixingLength 0.007;
        value uniform 1;
    }
    outlet
    {
        type inletOutlet;
        inletValue uniform 14.855;
        value uniform 14.855;
    }
    upperWall
    {
        type        epsilonWallFunction;
        value       uniform 14.855;
    }
    lowerWall
    {
        type        epsilonWallFunction;
        value       uniform 14.855;
    }
    frontAndBack
    {
        type        empty;
    }
}
```

３．境界条件：　比散逸率（その１）　ω

```
dimensions [ 0 0 -1 0 0 0 0 ];

internalField uniform 440.2;

boundaryField
{
   inlet
   {
      type          fixedValue;
      value          uniform 440.2;
   }
   outlet
   {
      type          zeroGradient;
   }

   upperWall
   {
      type          omegaWallFunction;
      value          uniform 440.2;
   }

   lowerWall
   {
      type          omegaWallFunction;
      value          uniform 440.2;
   }
   frontAndBack
   {
      type          empty;
   }
}
```

３．境界条件：　比散逸率（その２）　ω

```
dimensions [ 0 0 -1 0 0 0 0 ];

internalField uniform 440.2;

boundaryField
{
   inlet
   {
      type turbulentMixingLengthFrequencyInlet;
      mixingLength 0.007;
      k k;            // turbulent k field
      value uniform 440.2;
   }
   outlet
   {
      type inletOutlet;
      inletValue uniform 440.2;
      value uniform 440.2;
   }
   upperWall
   {
      type          omegaWallFunction;
      value          uniform 440.2;
   }
   lowerWall
   {
      type          omegaWallFunction;
      value          uniform 440.2;
   }
   frontAndBack
   {
      type          empty;
   }
}
```



4.数値スキーム　fvSchemes（1）

```
ddtSchemes
{
   default        steadyState;
}

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

divSchemes
{
   default        none;
   div(phi,U)     Gauss upwind;
   div(phi,k)     Gauss upwind;
   div(phi,epsilon) Gauss upwind;
   div(phi,R)     Gauss upwind;
   div(R)         Gauss linear;
   div(phi,nuTilda) Gauss upwind;
   div((nuEff*dev(grad(U).T()))) Gauss linear;
}
```

移流項

風上差分

```
laplacianSchemes
{
   default        none;
   laplacian(nuEff,U) Gauss linear corrected;
   laplacian((1|A(U)),p) Gauss linear corrected;
   laplacian(DkEff,k) Gauss linear corrected;
   laplacian(DepsilonEff,epsilon) Gauss linear corrected;
   laplacian(DREff,R) Gauss linear corrected;
   laplacian(DnuTildaEff,nuTilda) Gauss linear corrected;
}

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

snGradSchemes
{
   default        corrected;
}

fluxRequired
{
   default        no;
   p              ;
}
```

4.数値スキーム　fvSchemes（2）

```
ddtSchemes
{
    default steadyState;
}

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

divSchemes
{
    default        none;
    div(phi,U)     Gauss linear;
    div(phi,k)     Gauss linear;
    div(phi,epsilon) Gauss linear;
    div(phi,omega)  Gauss linear;
    div((nuEff*dev(grad(U).T()))) Gauss linear;
    div(phi,R)     Gauss linear;
    div(R)         Gauss linear;
    div(phi,nuTilda) Gauss linear;
}
```

移流項

中心差分

```
laplacianSchemes
{
    default        Gauss linear corrected;
//  default        Gauss linear limited 0.5;
//  default        Gauss linear limited 0.333;
}

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

snGradSchemes
{
    default        corrected;
}

fluxRequired
{
    default        no;
    p;
}
```

4.数値スキーム　fvSchemes（3）

```
ddtSchemes
{
    default steadyState;
}

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

divSchemes
{
    default        none;
    div(phi,U)     Gauss limitedLinearV 1;
    div(phi,k)     Gauss limitedLinear 1;
    div(phi,epsilon) Gauss limitedLinear 1;
    div(phi,omega)  Gauss limitedLinear 1;
    div((nuEff*dev(grad(U).T()))) Gauss linear;
    div(phi,R)     Gauss limitedLinear 1;
    div(R)         Gauss limitedLinear 1;
    div(phi,nuTilda) Gauss limitedLinear 1;
}
```

移流項

TVD法

```
laplacianSchemes
{
    default        Gauss linear corrected;
//  default        Gauss linear limited 0.5;
//  default        Gauss linear limited 0.333;
}

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

snGradSchemes
{
    default        corrected;
}

fluxRequired
{
    default        no;
    p;
}
```

５．解法とアルゴリズム制御　fvSolution（１）

```
solvers
{
    p
    {
        solver          PCG;
        preconditioner  DIC;
        tolerance       1e-06;
        relTol          0.01;
    }

    U
    {
        solver          PBiCG;
        preconditioner  DILU;
        tolerance       1e-05;
        relTol          0.1;
    }

    k
    {
        solver          PBiCG;
        preconditioner  DILU;
        tolerance       1e-05;
        relTol          0.1;
    }

    epsilon
    {
        solver          PBiCG;
        preconditioner  DILU;
        tolerance       1e-05;
        relTol          0.1;
    }

    R
    {
        solver          PBiCG;
        preconditioner  DILU;
        tolerance       1e-05;
        relTol          0.1;
    }

    nuTilda
    {
        solver          PBiCG;
        preconditioner  DILU;
        tolerance       1e-05;
        relTol          0.1;
    }
}

SIMPLE
{
    nNonOrthogonalCorrectors 0;
}

relaxationFactors
{
    p           0.3;
    U           0.7;
    k           0.7;
    epsilon     0.7;
    R           0.7;
    nuTilda     0.7;
}
```

Simple法ではpとUの緩和係数を足すと1が最適らしいが実際にもそうか？

初期条件付き共役勾配法

設定は，incompressible/simpleFoam/pitzDaily/systemを利用

５．解法とアルゴリズム制御　fvSolution（２）

```
solvers
{
   p
   {
      solver          GAMG;
      tolerance       1e-7;
      relTol          0.1;
      smoother        GaussSeidel;
      nPreSweeps      0;
      nPostSweeps     2;
      cacheAgglomeration on;
      agglomerator    faceAreaPair;
      nCellsInCoarsestLevel 10;
      mergeLevels     1;
   };

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

```
SIMPLE
{
   nNonOrthogonalCorrectors 0;
}

relaxationFactors
{
   p          0.3;
   U          0.7;
   k          0.7;
   epsilon    0.7;
   R          0.7;
   nuTilda    0.7;
}
```

GAMG:汎用幾何学的代数マルチグリッド

smoothSolver：スムーサを使ったソルバ

設定は，incompressible/simpleFoam/motorBike/systemを参照し一部変更

6. 計算設定　U,p,k(その1)，ε(その1)，ω(その1)

No	RANS モデル	時間(2000)	fvSchemes(U,k, ε ,ω)	fvSolution	緩和係数p,U,k,epsolin,omega
1	kEpsilon	147.7	upwind	PCG,PBiCG	0.3,0.7,0.7,0.7,0.7
2	kEpsilon	124.22	linear	PCG,PBiCG	0.3,0.7,0.7,0.7,0.7
3	kEpsilon	148.01	TVD	PCG,PBiCG	0.3,0.7,0.7,0.7,0.7
4	kEpsilon	88.9	upwind	GAMG,smoothSolver	0.3,0.7,0.7,0.7,0.7
5	kEpsilon	エラー	linear	GAMG,smoothSolver	0.3,0.7,0.7,0.7,0.7
6	kEpsilon	88.98	TVD	GAMG,smoothSolver	0.3,0.7,0.7,0.7,0.7
7	kOmegaSST	125.94	upwind	PCG,PBiCG	0.3,0.7,0.7,0.7,0.7
8	kOmegaSST	エラー	linear	PCG,PBiCG	0.3,0.7,0.7,0.7,0.7
9	kOmegaSST	エラー	TVD	PCG,PBiCG	0.3,0.7,0.7,0.7,0.7
10	kOmegaSST	95.93	upwind	GAMG,smoothSolver	0.3,0.7,0.7,0.7,0.7
11	kOmegaSST	92.12	linear	GAMG,smoothSolver	0.3,0.7,0.7,0.7,0.7
12	kOmegaSST	96.57	TVD	GAMG,smoothSolver	0.3,0.7,0.7,0.7,0.7
13	realizableKE	215.45	upwind	PCG,PBiCG	0.3,0.7,0.7,0.7,0.7
14	realizableKE	149.87	linear	PCG,PBiCG	0.3,0.7,0.7,0.7,0.7
15	realizableKE	163	TVD	PCG,PBiCG	0.3,0.7,0.7,0.7,0.7
16	realizableKE	93.87	upwind	GAMG,smoothSolver	0.3,0.7,0.7,0.7,0.7
17	realizableKE	84.85	linear	GAMG,smoothSolver	0.3,0.7,0.7,0.7,0.7
18	realizableKE	93.19	TVD	GAMG,smoothSolver	0.3,0.7,0.7,0.7,0.7

6. 計算設定　U,p,k（その2），ε（その2），ω（その2）

No	RANS モデル	時間(2000)	fvSchemes(U,k, ε ,ω)	fvSolution	緩和係数p,U,k,epsolin,omega
19	kEpsilon	148.29	upwind	PCG,PBiCG	0.3,0.7,0.7,0.7,0.7
20	kEpsilon	123.6	linear	PCG,PBiCG	0.3,0.7,0.7,0.7,0.7
21	kEpsilon	148.76	TVD	PCG,PBiCG	0.3,0.7,0.7,0.7,0.7
22	kEpsilon	89.27	upwind	GAMG,smoothSolver	0.3,0.7,0.7,0.7,0.7
23	kEpsilon	エラー	linear	GAMG,smoothSolver	0.3,0.7,0.7,0.7,0.7
24	kEpsilon	90.68	TVD	GAMG,smoothSolver	0.3,0.7,0.7,0.7,0.7
25	kOmegaSST	133.05	upwind	PCG,PBiCG	0.3,0.7,0.7,0.7,0.7
26	kOmegaSST	エラー	linear	PCG,PBiCG	0.3,0.7,0.7,0.7,0.7
27	kOmegaSST	エラー	TVD	PCG,PBiCG	0.3,0.7,0.7,0.7,0.7
28	kOmegaSST	97.18	upwind	GAMG,smoothSolver	0.3,0.7,0.7,0.7,0.7
29	kOmegaSST	88.45	linear	GAMG,smoothSolver	0.3,0.7,0.7,0.7,0.7
30	kOmegaSST	93.93	TVD	GAMG,smoothSolver	0.3,0.7,0.7,0.7,0.7
31	realizableKE	162.85	upwind	PCG,PBiCG	0.3,0.7,0.7,0.7,0.7
32	realizableKE	148.41	linear	PCG,PBiCG	0.3,0.7,0.7,0.7,0.7
33	realizableKE	167.7	TVD	PCG,PBiCG	0.3,0.7,0.7,0.7,0.7
34	realizableKE	94.48	upwind	GAMG,smoothSolver	0.3,0.7,0.7,0.7,0.7
35	realizableKE	83.09	linear	GAMG,smoothSolver	0.3,0.7,0.7,0.7,0.7
36	realizableKE	93.04	TVD	GAMG,smoothSolver	0.3,0.7,0.7,0.7,0.7

7．y+の疑問

y+は30～100辺りが良いと言われるが，今回の計算ではどうなっているのかチェックしてみる

		upperWall			lowerWall		
No	RANS モデル	y+ min	y+max	y+average	y+ min	y+max	y+average
1	kEpsilon	5.602	14.790	10.189	0.722	26.163	15.685
2	kEpsilon	5.334	15.161	10.249	0.916	28.206	16.934
3	kEpsilon	5.527	15.027	10.224	0.644	26.179	15.731
4	kEpsilon	5.602	14.790	10.189	0.722	26.165	15.689
5	kEpsilon	–	–	–	–	–	–
6	kEpsilon	5.527	15.027	10.224	0.644	26.179	15.730
7	kOmegaSST	0.449	12.212	5.205	0.485	11.628	7.230
8	kOmegaSST	–	–	–	–	–	–
9	kOmegaSST	–	–	–	–	–	–
10	kOmegaSST	2.187	12.165	7.911	0.026	20.259	11.827
11	kOmegaSST	1.605	12.332	7.900	0.003	20.150	11.511
12	kOmegaSST	1.640	12.234	7.847	0.001	20.050	11.470
13	realizableKE	5.135	14.378	9.895	0.144	23.657	12.432
14	realizableKE	4.955	14.555	9.881	0.468	27.390	14.102
15	realizableKE	4.982	14.431	9.874	0.121	23.797	12.531
16	realizableKE	5.135	14.378	9.895	0.144	23.657	12.429
17	realizableKE	3.256	14.555	9.740	0.371	28.535	15.386
18	realizableKE	3.067	14.431	9.554	0.171	33.934	16.492

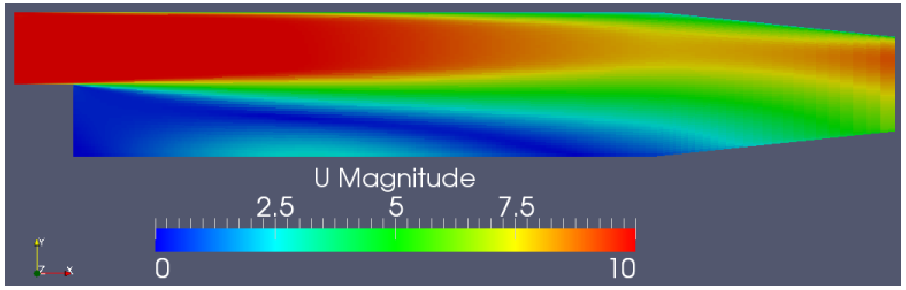
7．y+の疑問

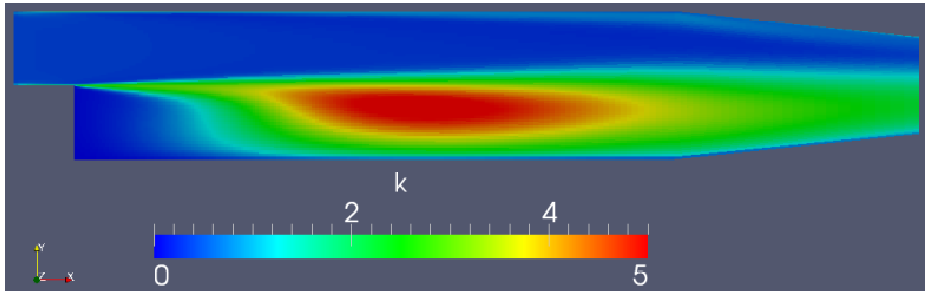
		upperWall			lowerWall		
No	RANS モデル	y+ min	y+max	y+average	y+ min	y+max	y+average
19	kEpsilon	5.602	14.790	10.189	0.722	26.163	15.685
20	kEpsilon	5.334	15.161	10.249	0.916	28.206	16.934
21	kEpsilon	5.527	15.027	10.224	0.644	26.179	15.731
22	kEpsilon	5.602	14.790	10.189	0.722	26.165	15.689
23	kEpsilon	–	–	–	–	–	
24	kEpsilon	5.527	15.027	10.224	0.644	26.179	15.730
25	kOmegaSST	0.449	12.212	5.205	0.485	11.628	7.230
26	kOmegaSST	–	–	–	–	–	–
27	kOmegaSST	–	–	–		–	–
28	kOmegaSST	2.187	12.165	7.911	0.026	20.259	11.827
29	kOmegaSST	1.605	12.332	7.900	0.003	20.150	11.511
30	kOmegaSST	1.640	12.234	7.847	0.001	20.050	11.470
31	realizableKE	5.135	14.378	9.895	0.144	23.657	12.432
32	realizableKE	4.955	14.555	9.881	0.468	27.390	14.102
33	realizableKE	4.982	14.431	9.874	0.121	23.797	12.531
34	realizableKE	5.135	14.378	9.895	0.144	23.657	12.429
35	realizableKE	3.256	14.555	9.740	0.371	28.535	15.386
36	realizableKE	3.067	14.431	9.554	0.171	33.934	16.492

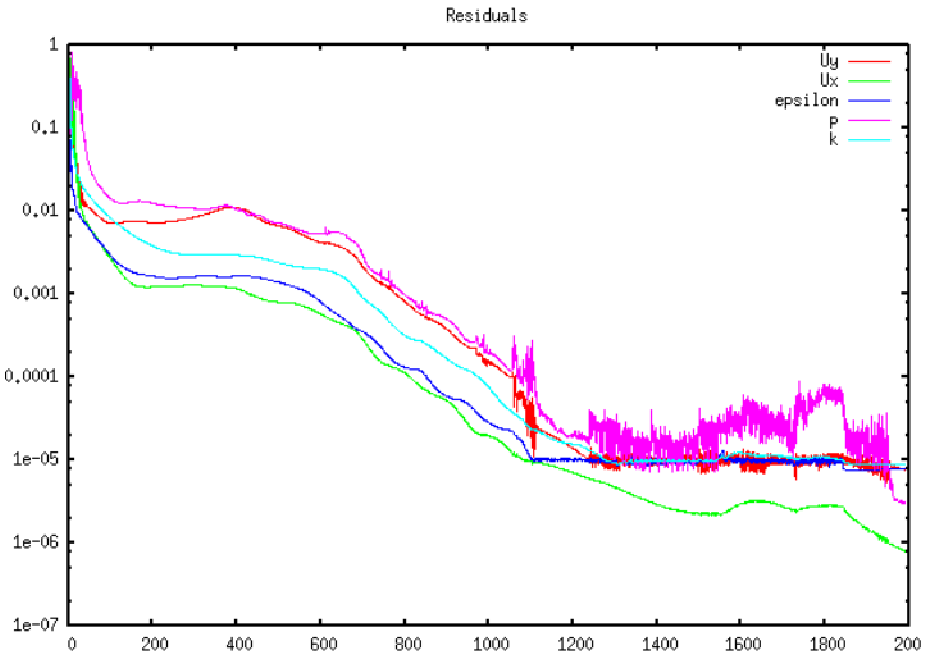
今回の事例は細かすぎるのか，推奨範囲より小さい。もう少しメッシュを粗くしても良いか？

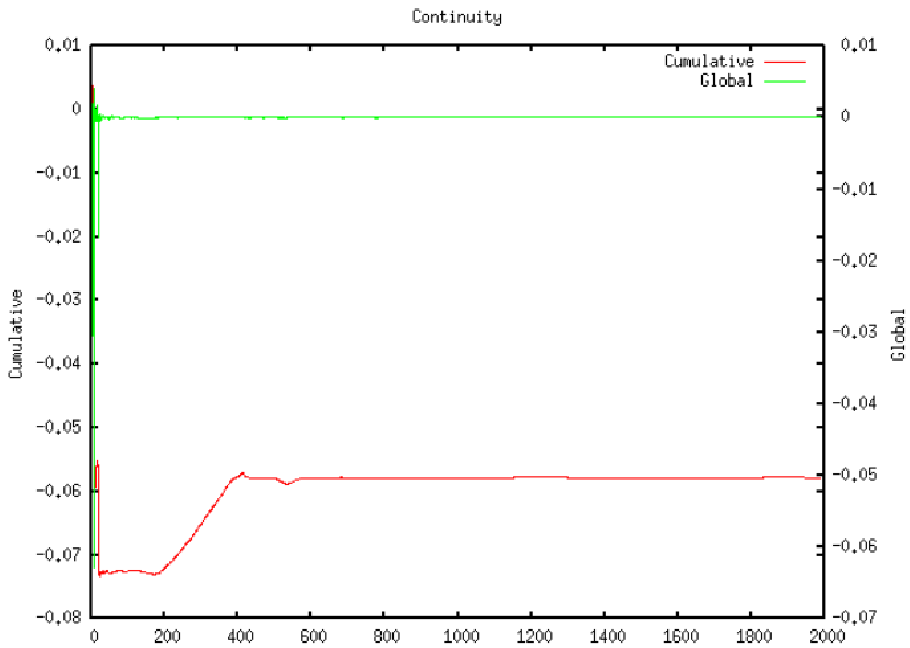
８. 計算結果の比較

No1　kEpsilon,upwind,PCG PBiCG





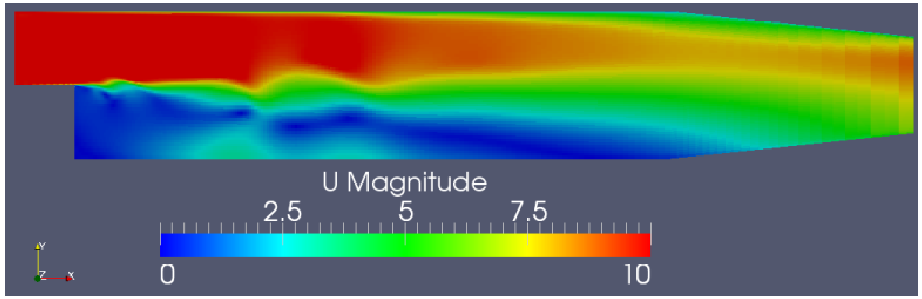


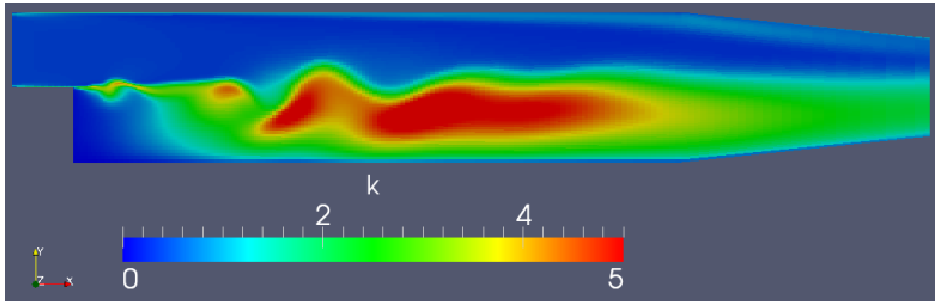


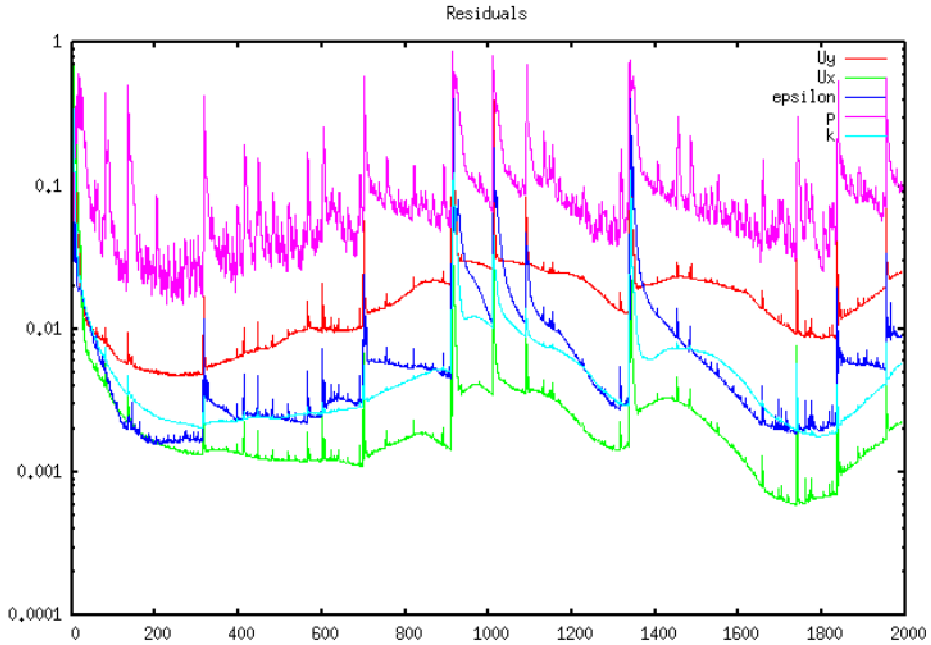
８．計算結果の比較

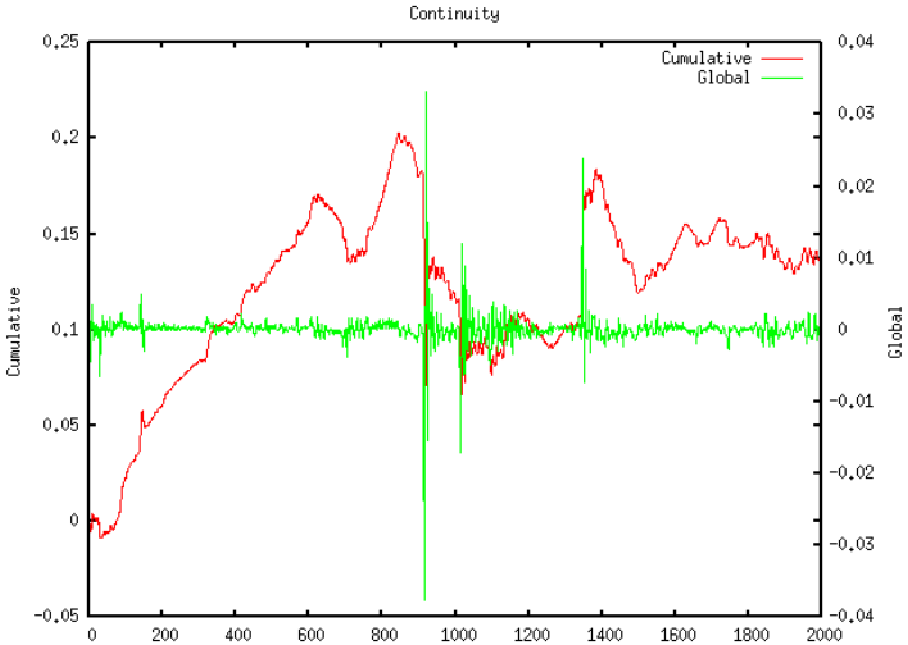
No2　kEpsilon,linear,PCG PBiCG

収束が安定しない



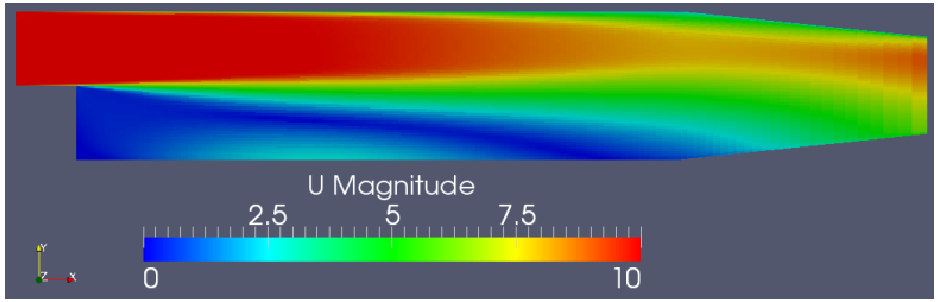


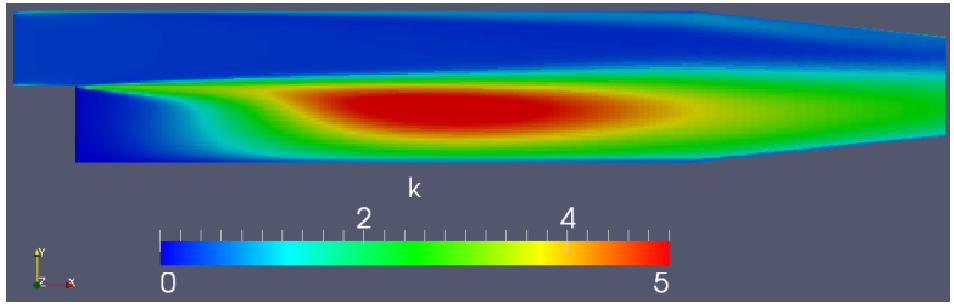


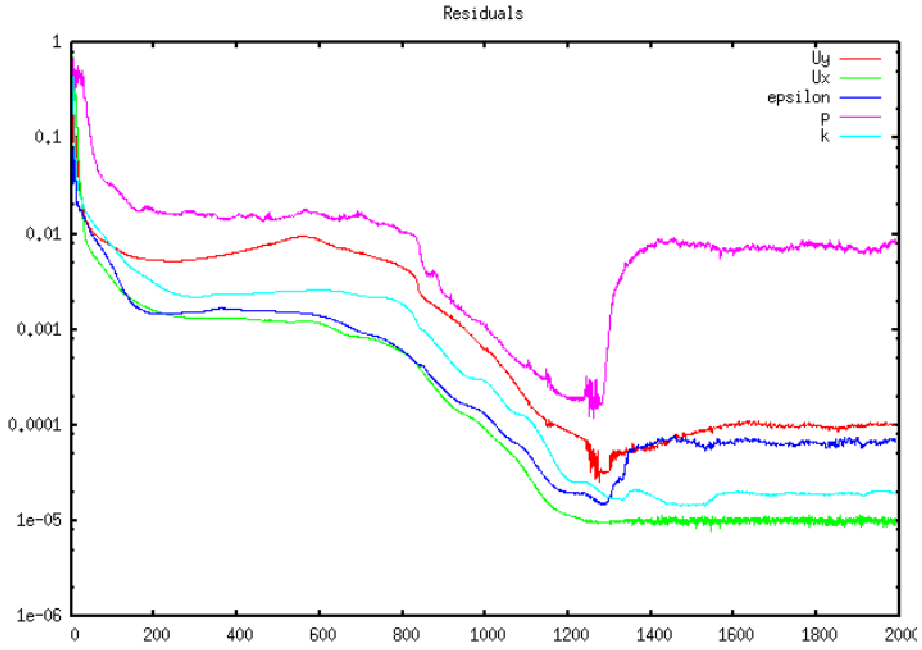


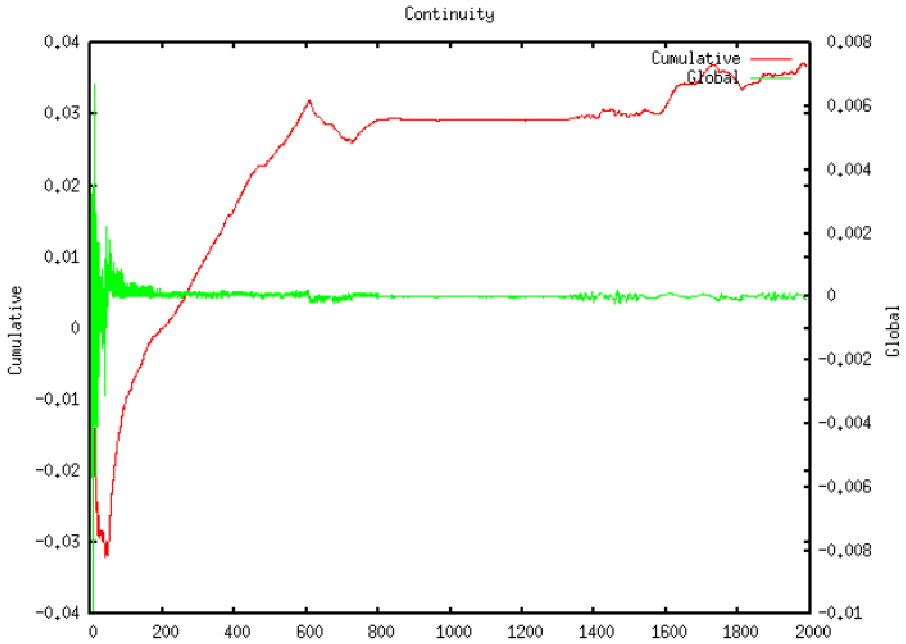
８. 計算結果の比較

No3　kEpsilon,TVD,PCG PBiCG



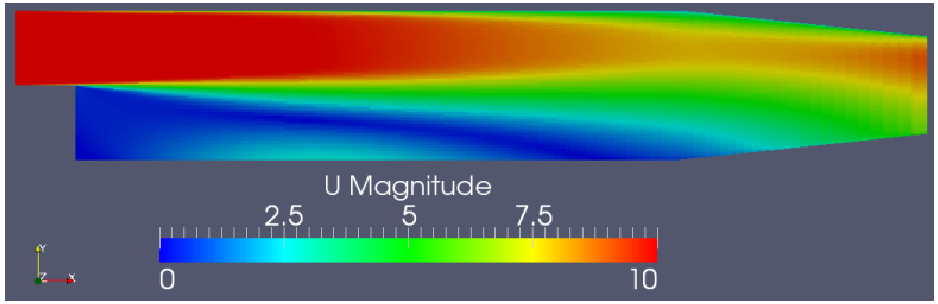


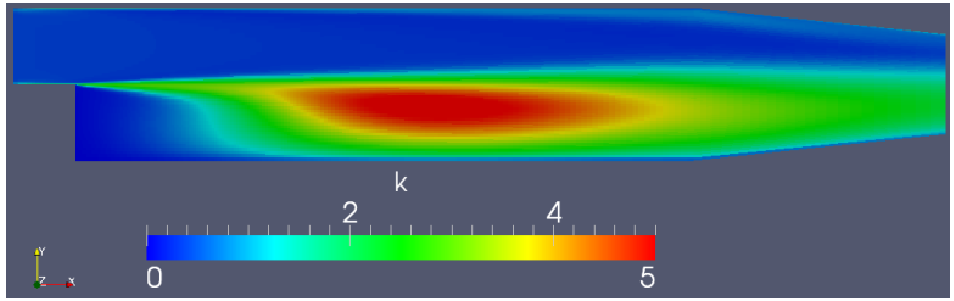


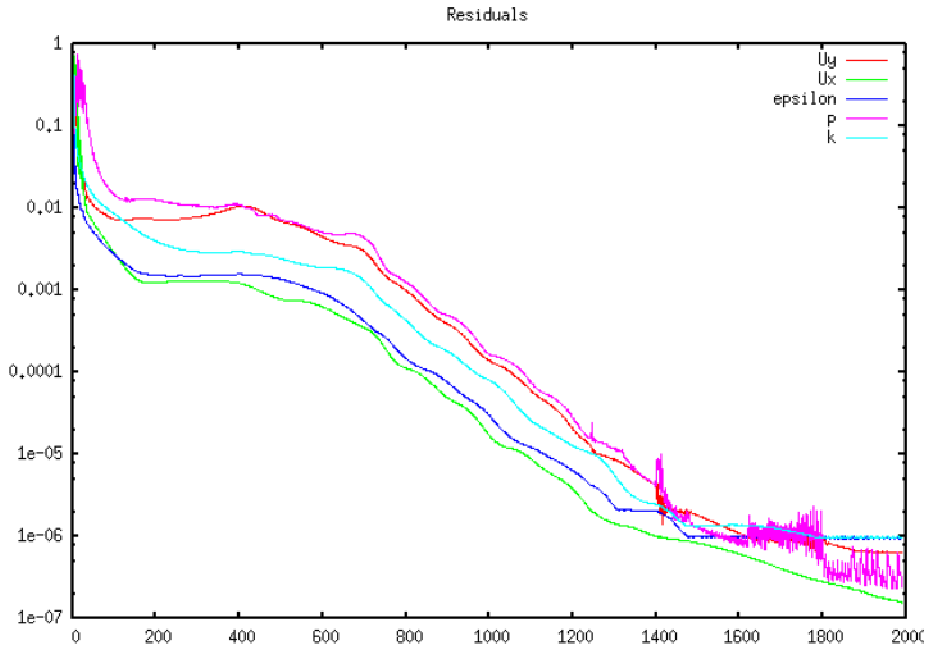


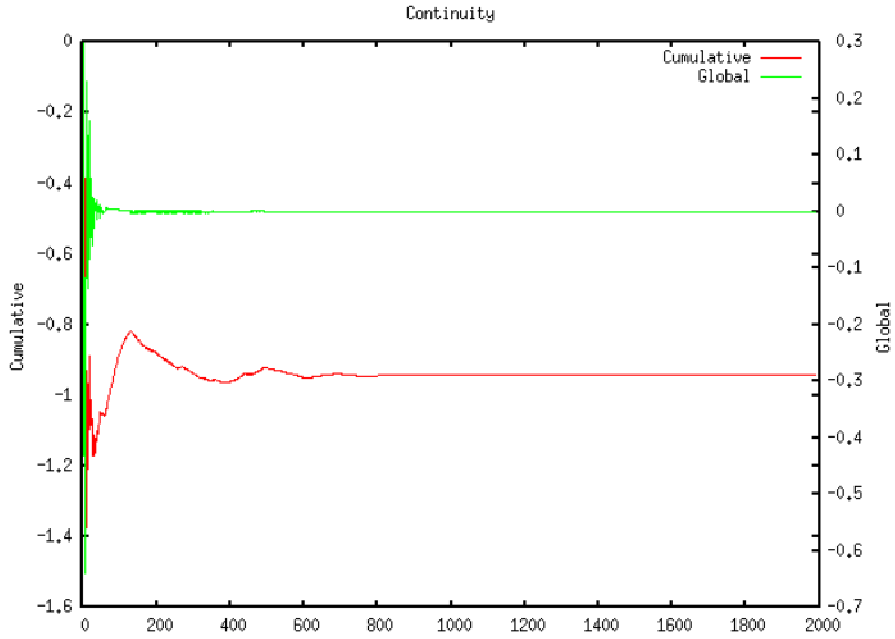
８. 計算結果の比較

No4　kEpsilon,upwind,GAMG smoothSolver









８．計算結果の比較

No5　kEpsilon,linear,GAMG smoothSolver

```
#0  Foam::error::printStack(Foam::Ostream&) in "/home/sakuramaru/OpenFOAM/OpenFOAM-
1.7.x/lib/linux64GccDPOpt/libOpenFOAM.so"
#1  Foam::sigFpe::sigFpeHandler(int) in "/home/sakuramaru/OpenFOAM/OpenFOAM-
1.7.x/lib/linux64GccDPOpt/libOpenFOAM.so"
#2   in "/lib/libc.so.6"
#3  Foam::PCG::solve(Foam::Field<double>&, Foam::Field<double> const&, unsigned char) const in
"/home/sakuramaru/OpenFOAM/OpenFOAM-1.7.x/lib/linux64GccDPOpt/libOpenFOAM.so"
#4  Foam::GAMGSolver::solveCoarsestLevel(Foam::Field<double>&, Foam::Field<double> const&) const in
"/home/sakuramaru/OpenFOAM/OpenFOAM-1.7.x/lib/linux64GccDPOpt/libOpenFOAM.so"
#5  Foam::GAMGSolver::Vcycle(Foam::PtrList<Foam::lduMatrix::smoother> const&, Foam::Field<double>&,
Foam::Field<double> const&, Foam::Field<double>&, Foam::Field<double>&, Foam::Field<double>&,
Foam::PtrList<Foam::Field<double> >&, Foam::PtrList<Foam::Field<double> >&, unsigned char) const in
"/home/sakuramaru/OpenFOAM/OpenFOAM-1.7.x/lib/linux64GccDPOpt/libOpenFOAM.so"
#6  Foam::GAMGSolver::solve(Foam::Field<double>&, Foam::Field<double> const&, unsigned char) const in
"/home/sakuramaru/OpenFOAM/OpenFOAM-1.7.x/lib/linux64GccDPOpt/libOpenFOAM.so"
#7  Foam::fvMatrix<double>::solve(Foam::dictionary const&) in "/home/sakuramaru/OpenFOAM/OpenFOAM-
1.7.x/lib/linux64GccDPOpt/libfiniteVolume.so"
#8
 in "/home/sakuramaru/OpenFOAM/OpenFOAM-1.7.x/applications/bin/linux64GccDPOpt/simpleFoam"
#9  __libc_start_main in "/lib/libc.so.6"
#10
 in "/home/sakuramaru/OpenFOAM/OpenFOAM-1.7.x/applications/bin/linux64GccDPOpt/simpleFoam"
Floating point exception
```

見方が良く分かりません。

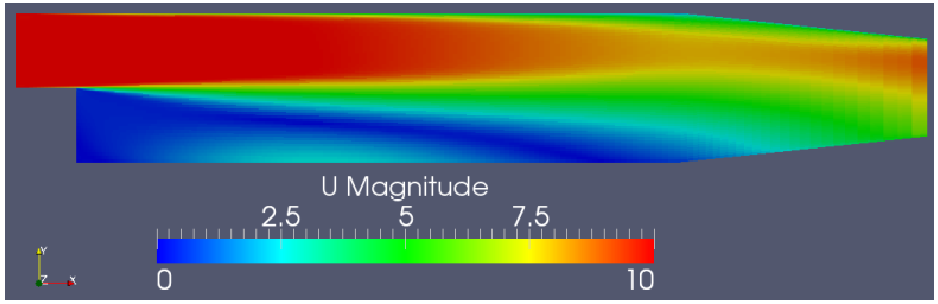
８. 計算結果の比較

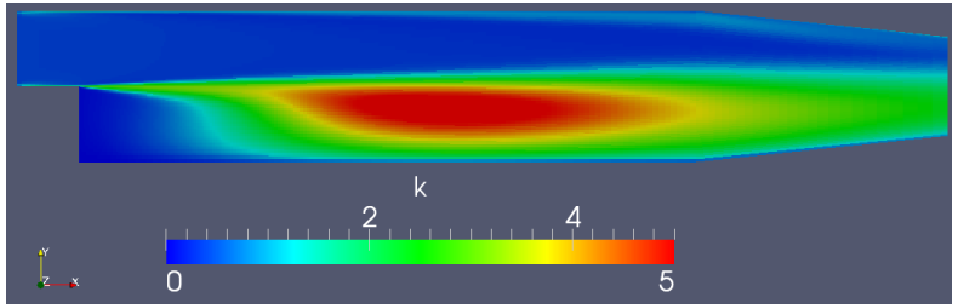
No5　kEpsilon,linear,GAMG smoothSolver

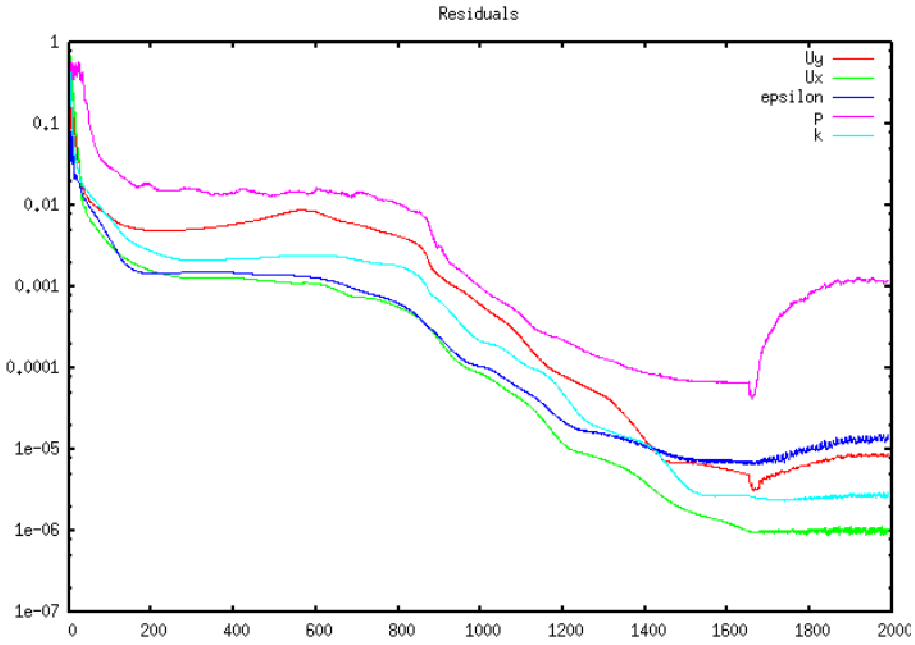
Time = 1030
smoothSolver: Solving for Ux, Initial residual = 0.764566, Final residual = 0.0397834, No Iterations 3
smoothSolver: Solving for Uy, Initial residual = 0.209879, Final residual = 0.0148312, No Iterations 4
GAMG: Solving for p, Initial residual = 1, Final residual = 0.0392114, No Iterations 3
time step continuity errors : sum local = 1.31161e+43, global = -1.22957e+42, cumulative = -1.22957e+42
smoothSolver: Solving for epsilon, Initial residual = 1, Final residual = 0.0862734, No Iterations 4
bounding epsilon, min: -3.38166e+70 max: 1.92608e+73 average: 1.61417e+69
smoothSolver: Solving for k, Initial residual = 1, Final residual = 0.041405, No Iterations 3
bounding k, min: -6.30268e+57 max: 5.32925e+58 average: 6.00644e+55
ExecutionTime = 41.42 s ClockTime = 42 s

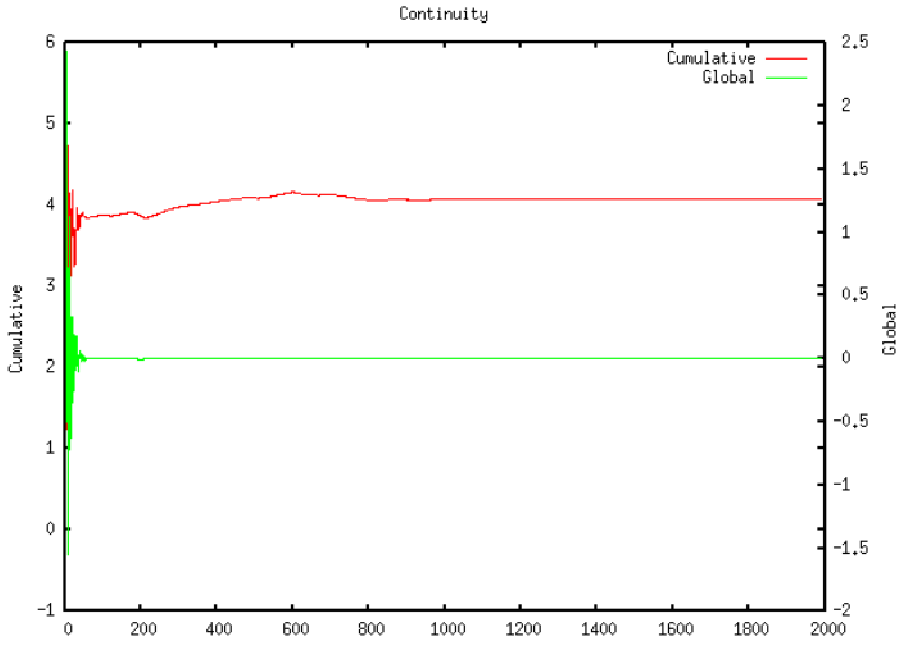
８. 計算結果の比較

No6　kEpsilon,TVD,GAMG smoothSolver





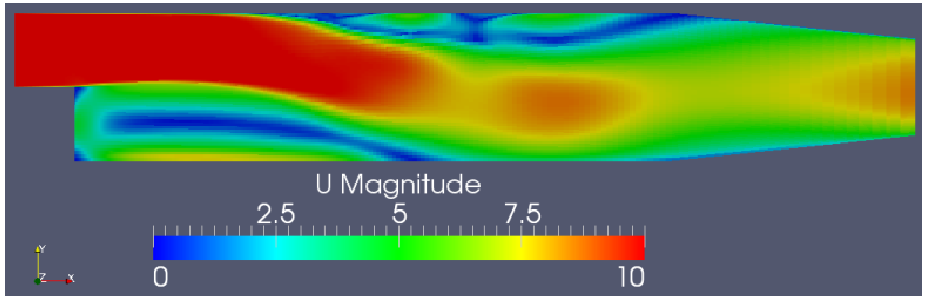


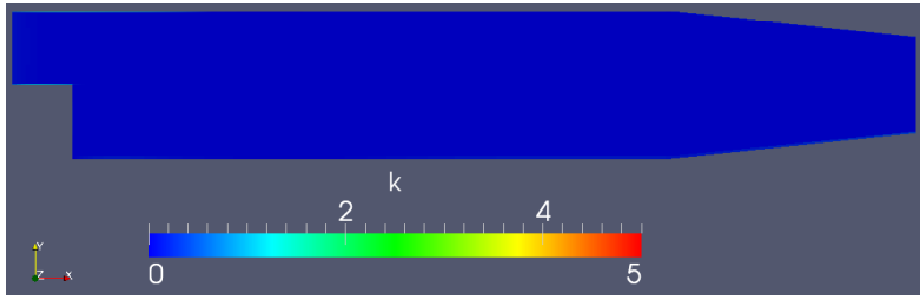


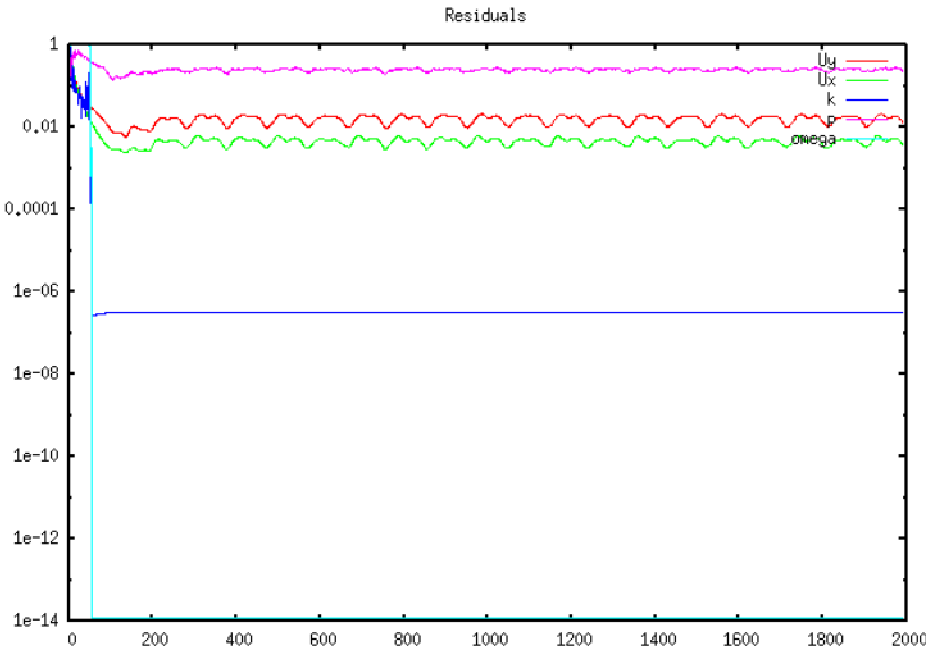
８．計算結果の比較

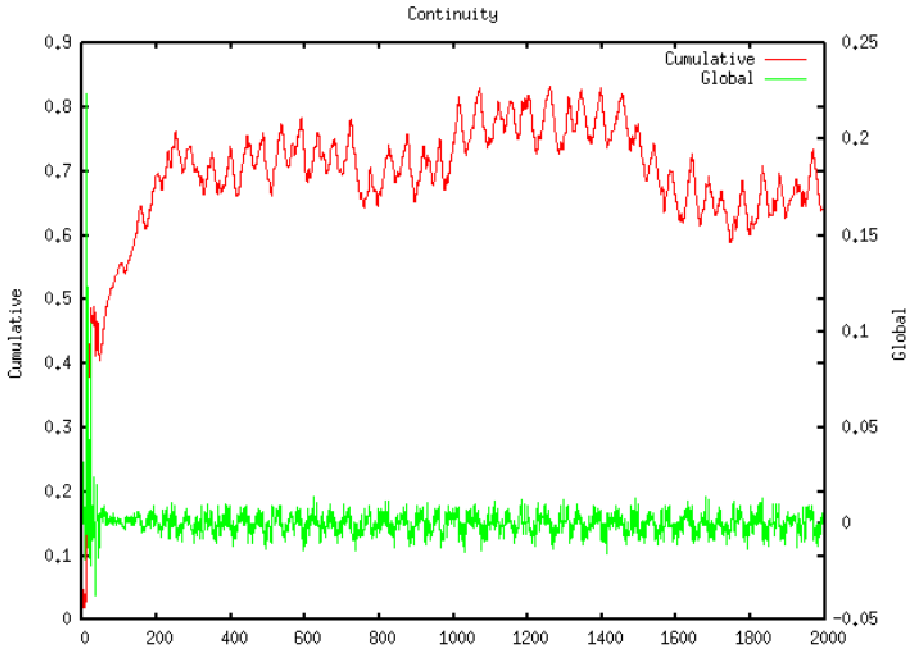
No7　kOmegaSST,upwind,PCG PBiCG

収束がおかしい









８．計算結果の比較

No8　kOmegaSST,linear,PCG PBiCG

```
#0  Foam::error::printStack(Foam::Ostream&) in "/home/sakuramaru/OpenFOAM/OpenFOAM-
1.7.x/lib/linux64GccDPOpt/libOpenFOAM.so"
#1  Foam::sigFpe::sigFpeHandler(int) in "/home/sakuramaru/OpenFOAM/OpenFOAM-
1.7.x/lib/linux64GccDPOpt/libOpenFOAM.so"
#2   in "/lib/libc.so.6"
#3  Foam::PBiCG::solve(Foam::Field<double>&, Foam::Field<double> const&, unsigned char) const in
"/home/sakuramaru/OpenFOAM/OpenFOAM-1.7.x/lib/linux64GccDPOpt/libOpenFOAM.so"
#4  Foam::fvMatrix<double>::solve(Foam::dictionary const&) in "/home/sakuramaru/OpenFOAM/OpenFOAM-
1.7.x/lib/linux64GccDPOpt/libfiniteVolume.so"
#5  Foam::lduMatrix::solverPerformance Foam::solve<double>(Foam::tmp<Foam::fvMatrix<double> > const&) in
"/home/sakuramaru/OpenFOAM/OpenFOAM-1.7.x/lib/linux64GccDPOpt/libincompressibleRASModels.so"
#6  Foam::incompressible::RASModels::kOmegaSST::correct() in "/home/sakuramaru/OpenFOAM/OpenFOAM-
1.7.x/lib/linux64GccDPOpt/libincompressibleRASModels.so"
#7
 in "/home/sakuramaru/OpenFOAM/OpenFOAM-1.7.x/applications/bin/linux64GccDPOpt/simpleFoam"
#8  __libc_start_main in "/lib/libc.so.6"
#9
 in "/home/sakuramaru/OpenFOAM/OpenFOAM-1.7.x/applications/bin/linux64GccDPOpt/simpleFoam"
Floating point exception
```

８．計算結果の比較

No8　kOmegaSST,linear,PCG PBiCG

```
Time = 13
DILUPBiCG:  Solving for Ux, Initial residual = 0.0882911, Final residual = 0.00176956, No Iterations 1
DILUPBiCG:  Solving for Uy, Initial residual = 0.0589643, Final residual = 9.40842e-05, No Iterations 2
DICPCG:  Solving for p, Initial residual = 0.293575, Final residual = 0.00284045, No Iterations 141
time step continuity errors : sum local = 1.02058, global = 0.0136782, cumulative = 0.236392
DILUPBiCG:  Solving for omega, Initial residual = 1, Final residual = 4.15499e-18, No Iterations 1
bounding omega, min: -8.83428e+108 max: 2.08912e+109 average: 1.05743e+105
DILUPBiCG:  Solving for k, Initial residual = 0.00493906, Final residual = 1.18318e-22, No Iterations 1
bounding k, min: -0.218238 max: 3.12828 average: 0.0177913
ExecutionTime = 2.19 s  ClockTime = 2 s
```

８．計算結果の比較

No9　kOmegaSST,TVD,PCG PBiCG

```
 #0  Foam::error::printStack(Foam::Ostream&) in "/home/sakuramaru/OpenFOAM/OpenFOAM-
1.7.x/lib/linux64GccDPOpt/libOpenFOAM.so"
#1  Foam::sigFpe::sigFpeHandler(int) in "/home/sakuramaru/OpenFOAM/OpenFOAM-
1.7.x/lib/linux64GccDPOpt/libOpenFOAM.so"
#2   in "/lib/libc.so.6"
#3  Foam::LimitedScheme<double, Foam::limitedLinearLimiter<Foam::NVDTVD>,
Foam::limitFuncs::magSqr>::limiter(Foam::GeometricField<double, Foam::fvPatchField, Foam::volMesh> const&) const in
"/home/sakuramaru/OpenFOAM/OpenFOAM-1.7.x/lib/linux64GccDPOpt/libfiniteVolume.so"
#4  Foam::limitedSurfaceInterpolationScheme<double>::weights(Foam::GeometricField<double, Foam::fvPatchField,
Foam::volMesh> const&) const in "/home/sakuramaru/OpenFOAM/OpenFOAM-1.7.x/lib/linux64GccDPOpt/libfiniteVolume.so"
#5  Foam::fv::gaussConvectionScheme<double>::fvmDiv(Foam::GeometricField<double, Foam::fvsPatchField,
Foam::surfaceMesh> const&, Foam::GeometricField<double, Foam::fvPatchField, Foam::volMesh>&) const in
"/home/sakuramaru/OpenFOAM/OpenFOAM-1.7.x/lib/linux64GccDPOpt/libfiniteVolume.so"
#6  Foam::tmp<Foam::fvMatrix<double> > Foam::fvm::div<double>(Foam::GeometricField<double, Foam::fvsPatchField,
Foam::surfaceMesh> const&, Foam::GeometricField<double, Foam::fvPatchField, Foam::volMesh>&, Foam::word const&) in
"/home/sakuramaru/OpenFOAM/OpenFOAM-1.7.x/lib/linux64GccDPOpt/libincompressibleRASModels.so"
#7  Foam::tmp<Foam::fvMatrix<double> > Foam::fvm::div<double>(Foam::GeometricField<double, Foam::fvsPatchField,
Foam::surfaceMesh> const&, Foam::GeometricField<double, Foam::fvPatchField, Foam::volMesh>&) in
"/home/sakuramaru/OpenFOAM/OpenFOAM-1.7.x/lib/linux64GccDPOpt/libincompressibleRASModels.so"
#8  Foam::incompressible::RASModels::kOmegaSST::correct() in "/home/sakuramaru/OpenFOAM/OpenFOAM-
1.7.x/lib/linux64GccDPOpt/libincompressibleRASModels.so"
#9
 in "/home/sakuramaru/OpenFOAM/OpenFOAM-1.7.x/applications/bin/linux64GccDPOpt/simpleFoam"
#10  __libc_start_main in "/lib/libc.so.6"
#11
 in "/home/sakuramaru/OpenFOAM/OpenFOAM-1.7.x/applications/bin/linux64GccDPOpt/simpleFoam"
Floating point exception
```

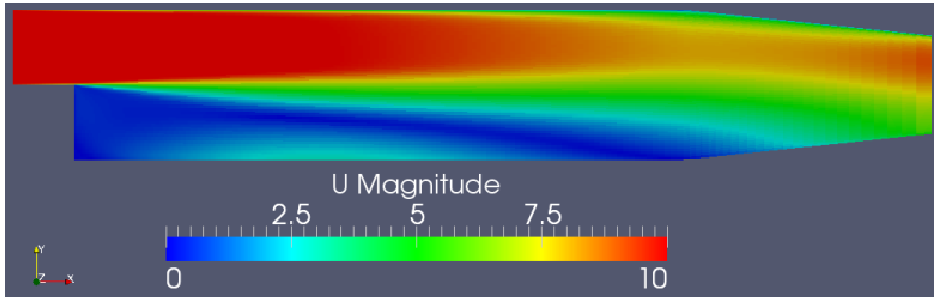
８. 計算結果の比較

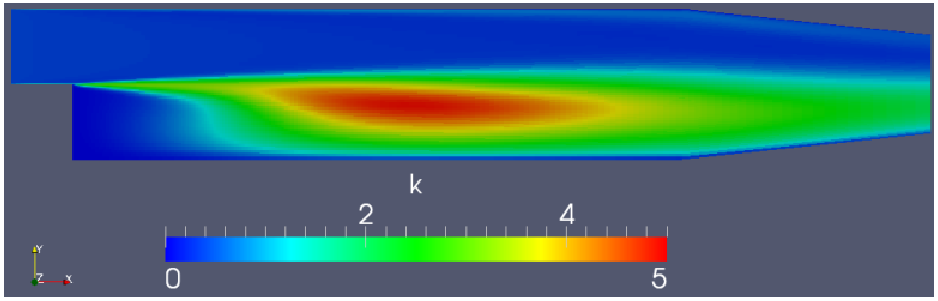
No9　kOmegaSST,TVD,PCG PBiCG

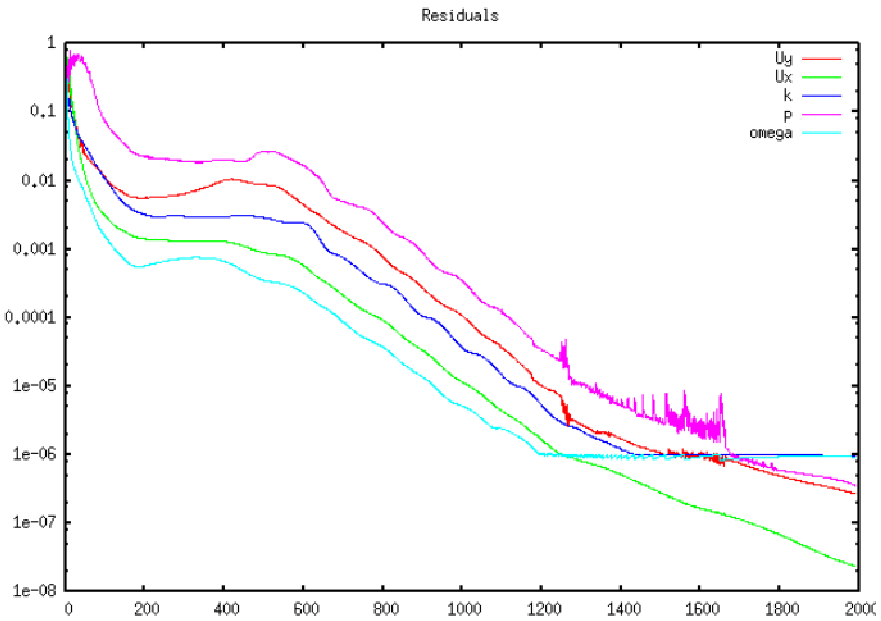
Time = 3
DILUPBiCG: Solving for Ux, Initial residual = 0.532708, Final residual = 0.033915, No Iterations 1
DILUPBiCG: Solving for Uy, Initial residual = 0.195258, Final residual = 0.0083128, No Iterations 1
DICPCG: Solving for p, Initial residual = 0.158018, Final residual = 0.00147366, No Iterations 141
time step continuity errors : sum local = 2.20071, global = 0.0175231, cumulative = 0.0482703
DILUPBiCG: Solving for omega, Initial residual = 1, Final residual = 0.0634549, No Iterations 26
bounding omega, min: -1.61642e+19 max: 1.62096e+21 average: 2.09599e+19
DILUPBiCG: Solving for k, Initial residual = 0.933999, Final residual = 7.4627e-17, No Iterations 1
ExecutionTime = 0.53 s ClockTime = 1 s

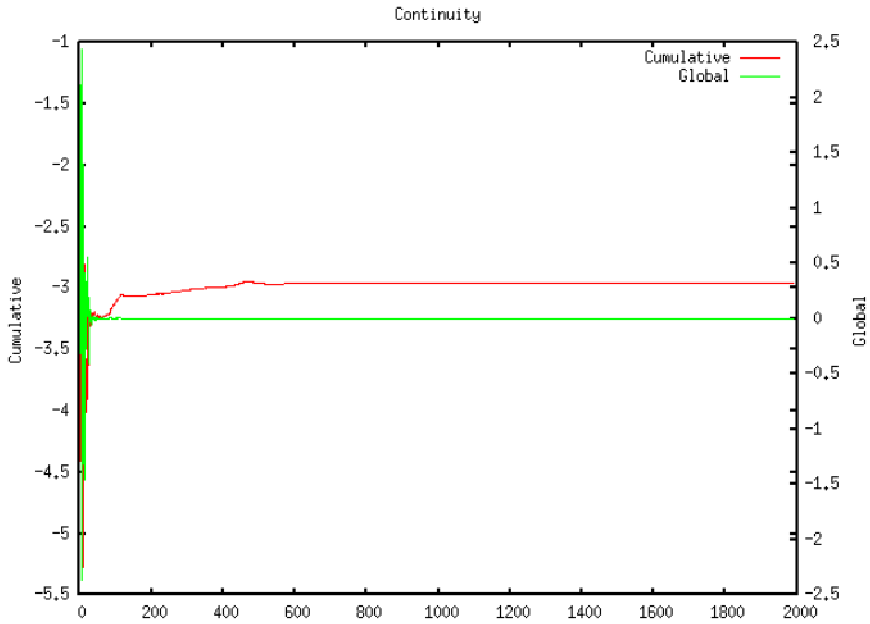
８. 計算結果の比較

No10　kOmegaSST,upwind,GAMG smoothSolver



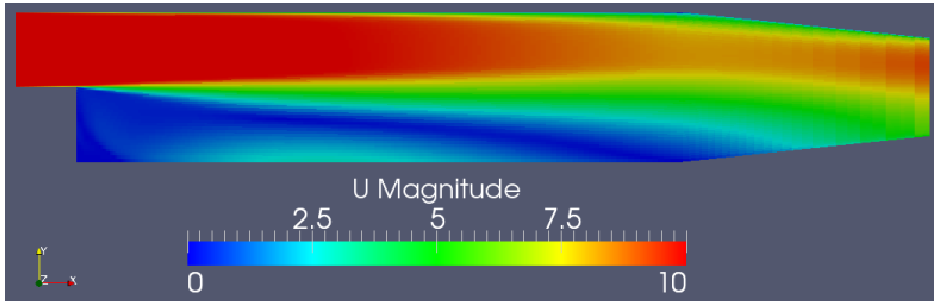


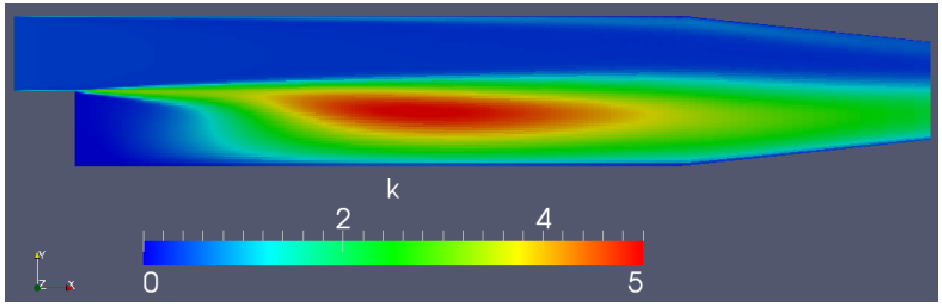


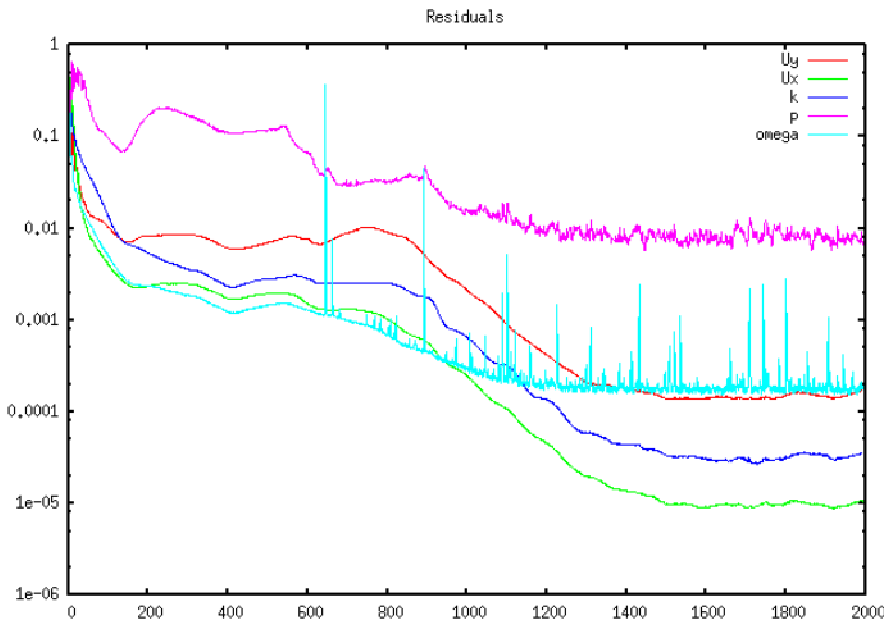


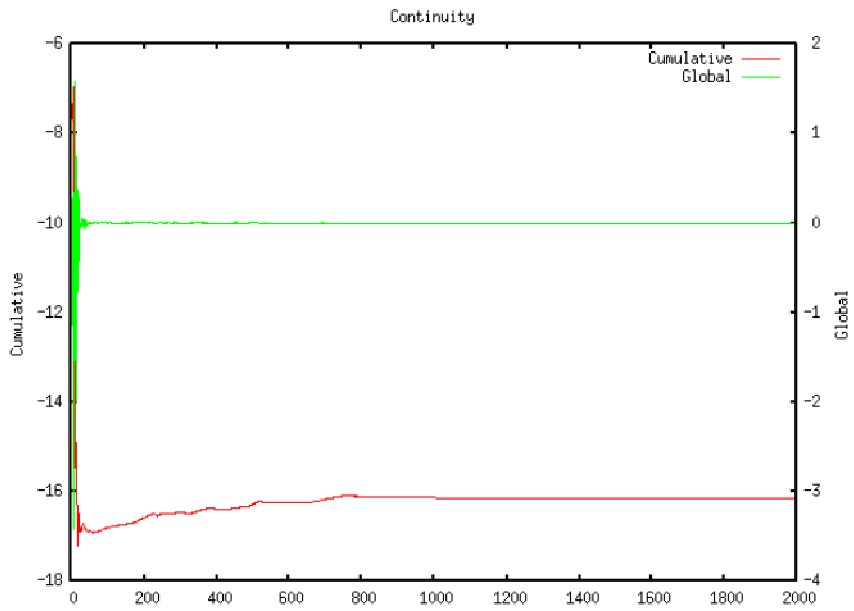
８. 計算結果の比較

No11　kOmegaSST,linear,GAMG smoothSolver



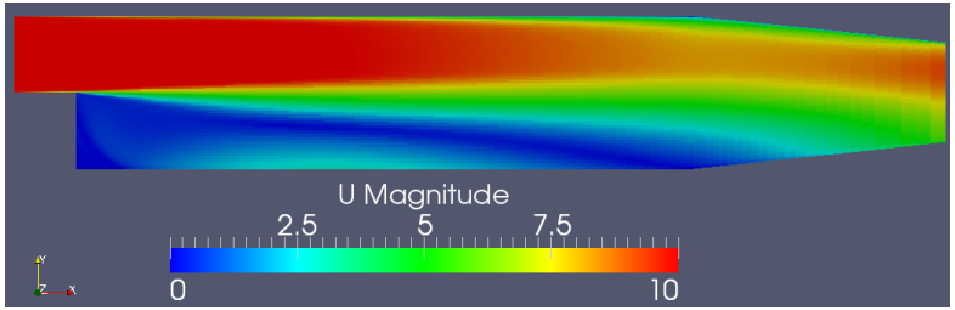


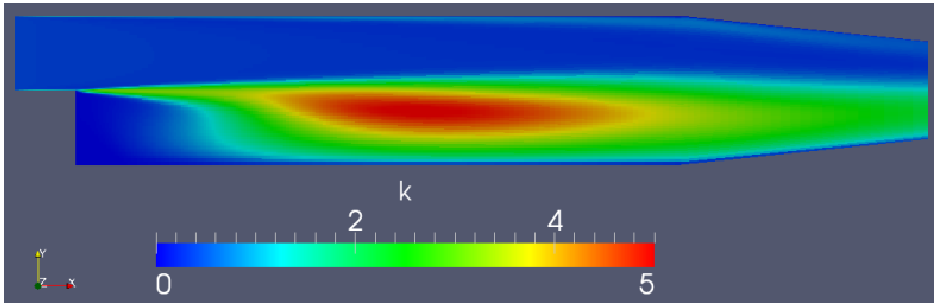


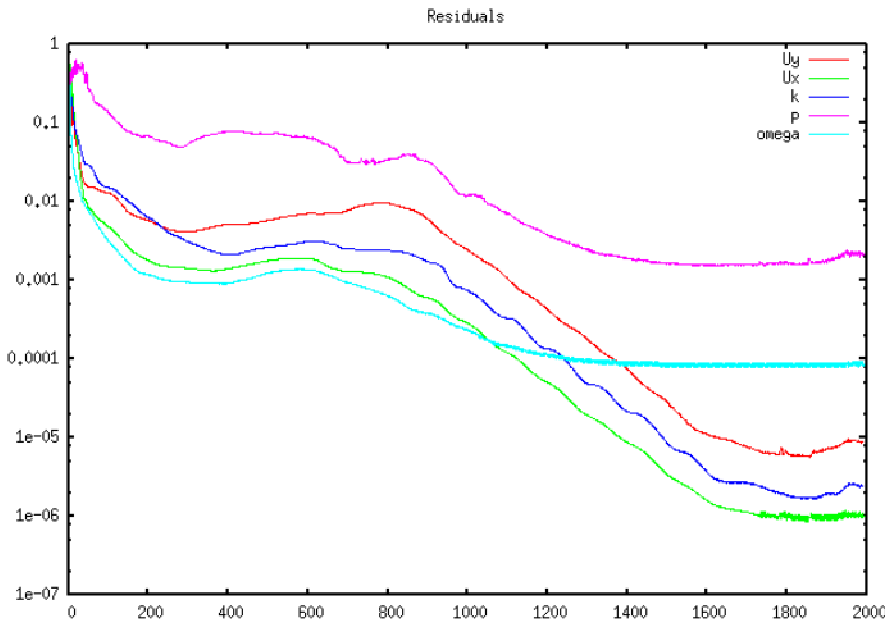


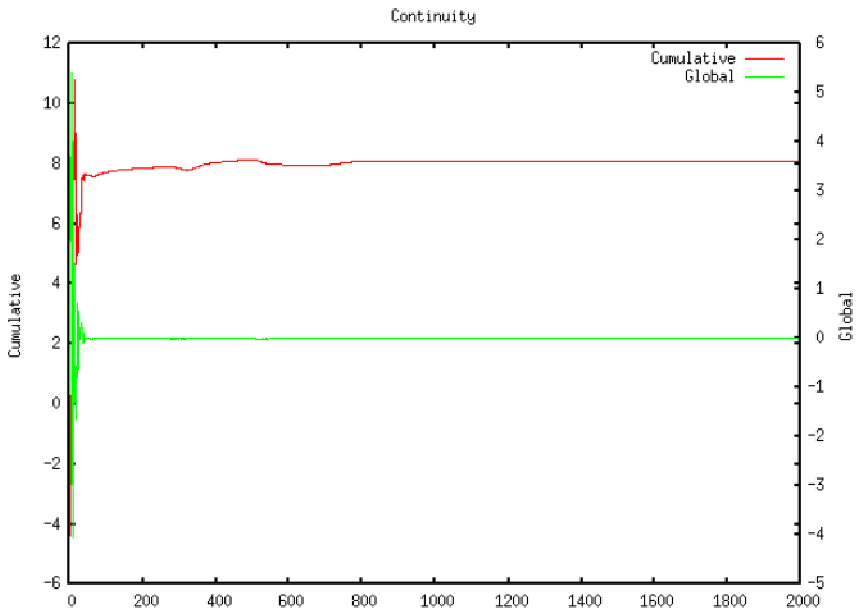
８. 計算結果の比較

No12　kOmegaSST,TVD,GAMG smoothSolver



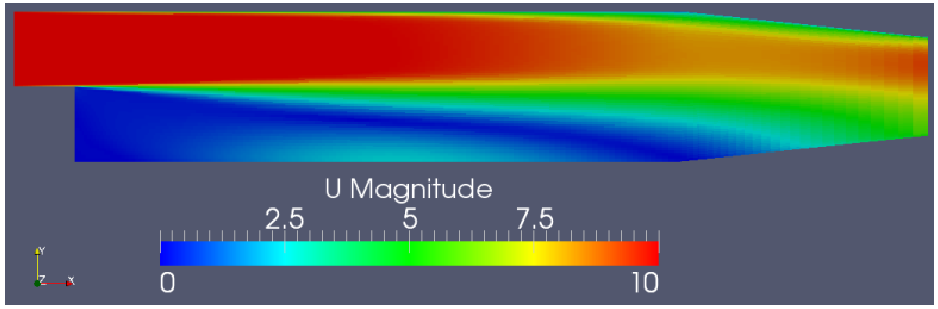


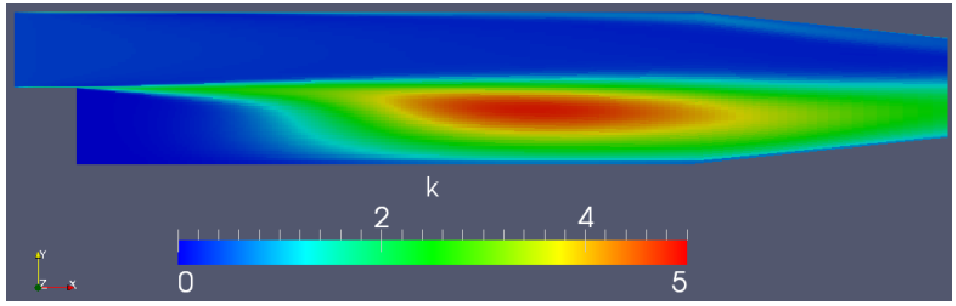


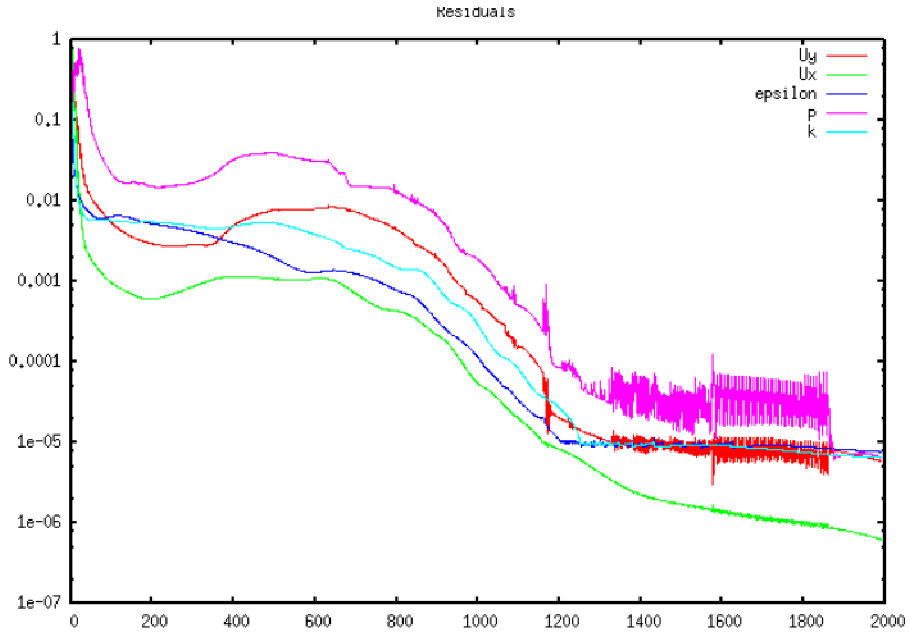


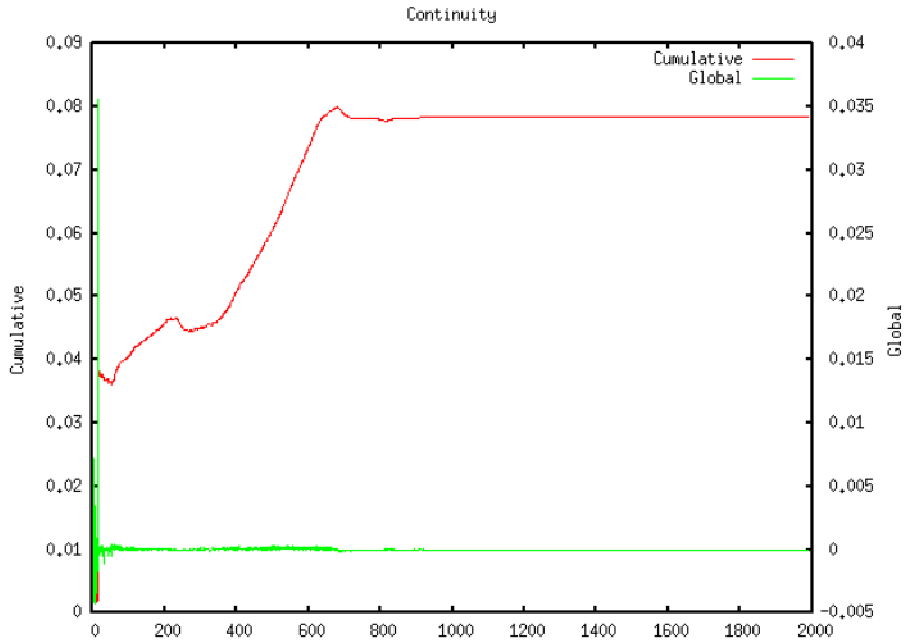
８. 計算結果の比較

No13　realizableKE,upwind,PCG PBiCG





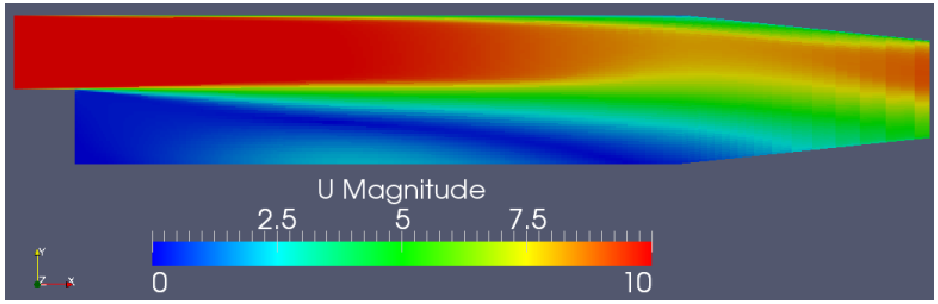


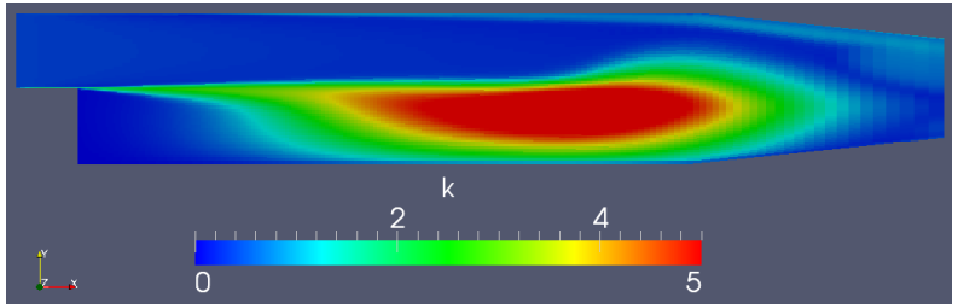


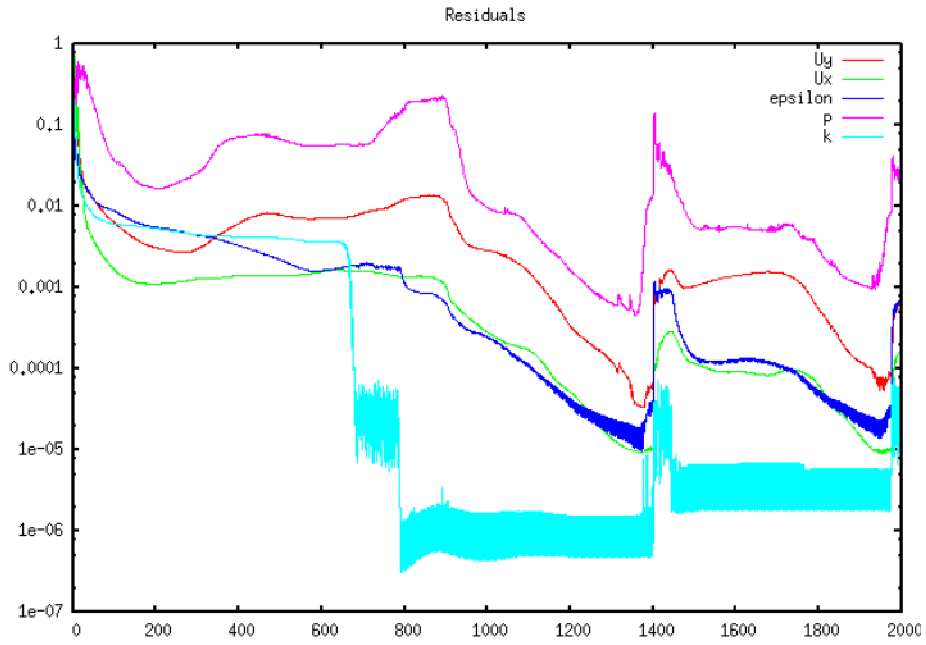
８．計算結果の比較

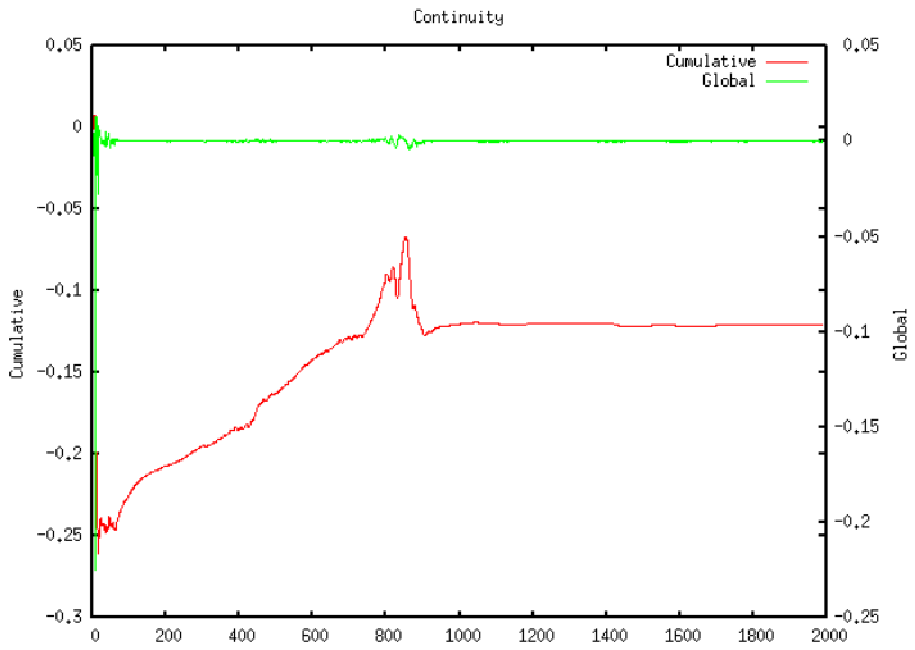
No14　realizableKE,linear,PCG PBiCG

収束が安定しない？



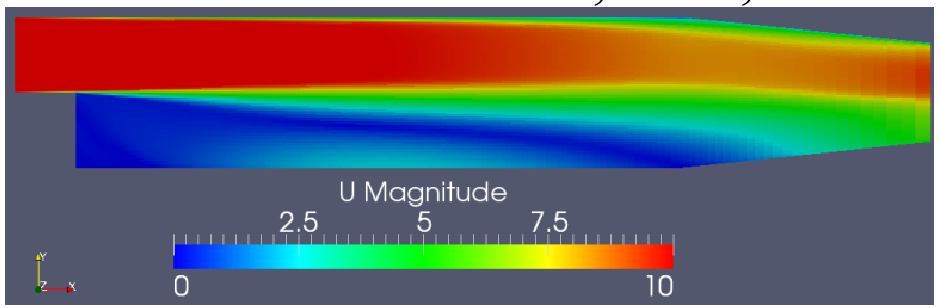


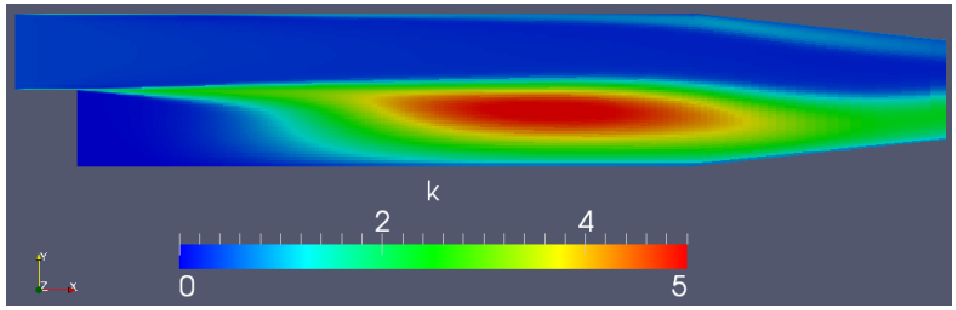


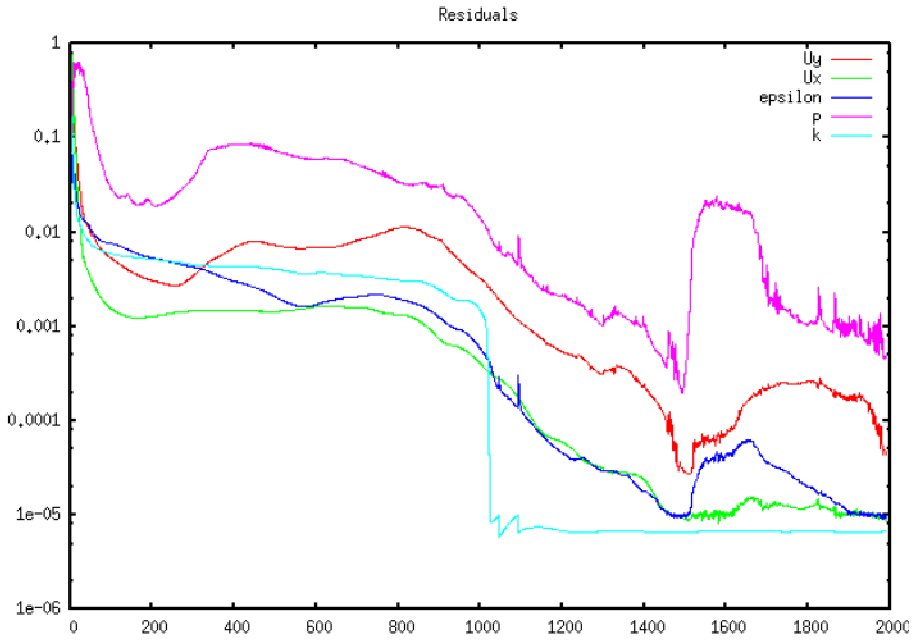


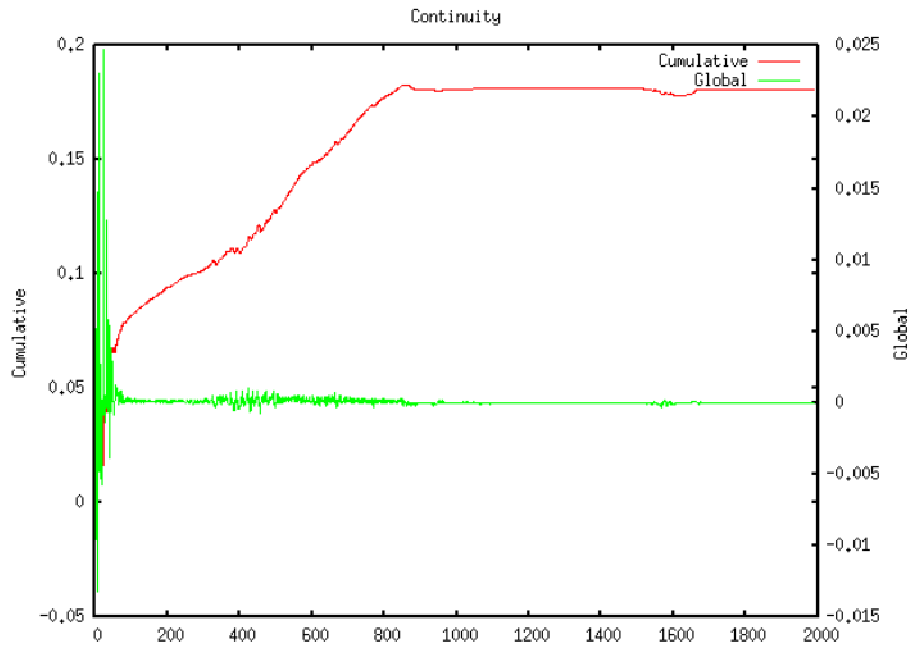
８. 計算結果の比較

No15　realizableKE,TVD,PCG PBiCG



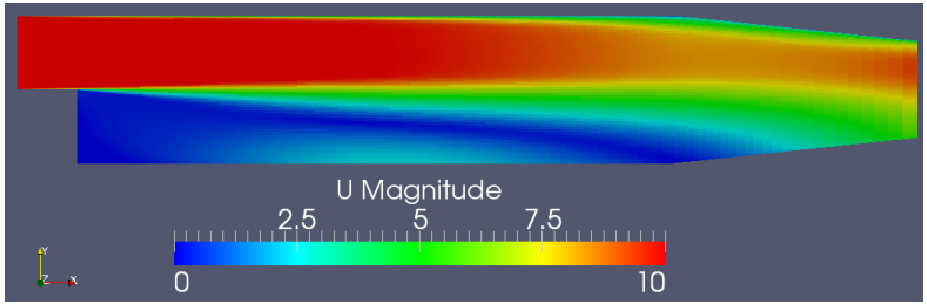


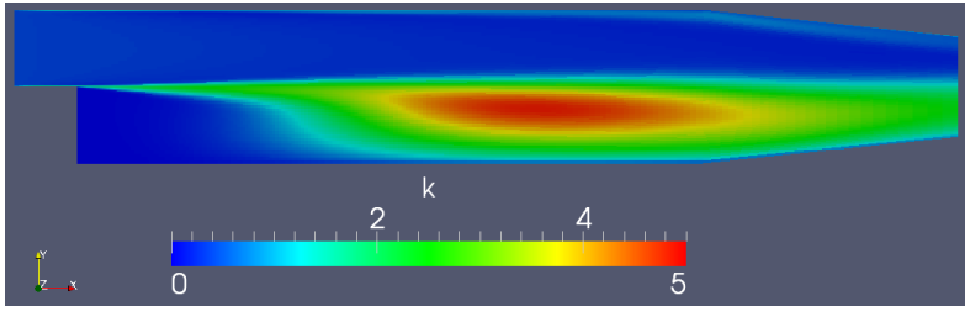


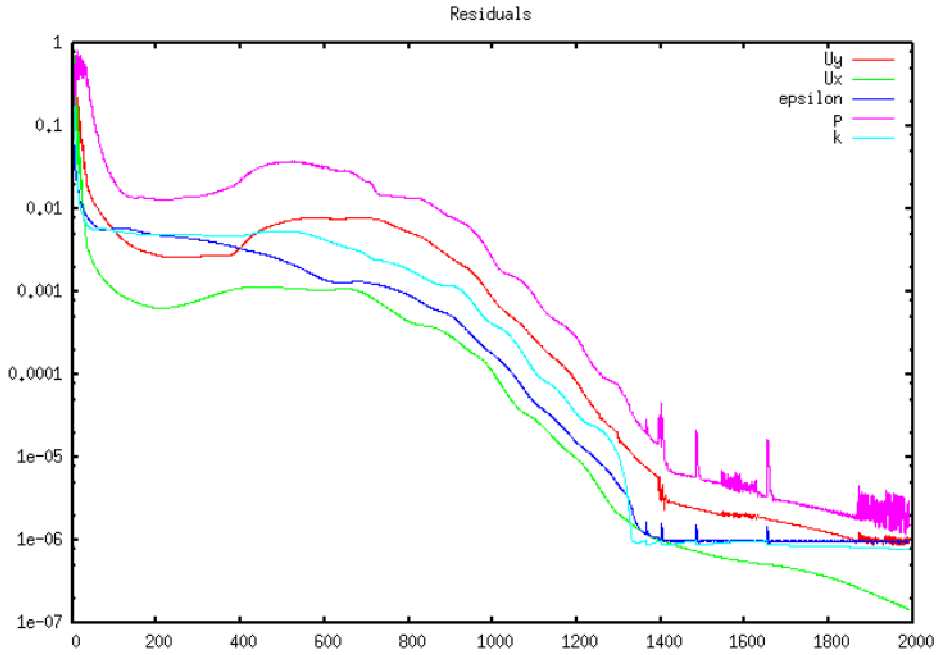


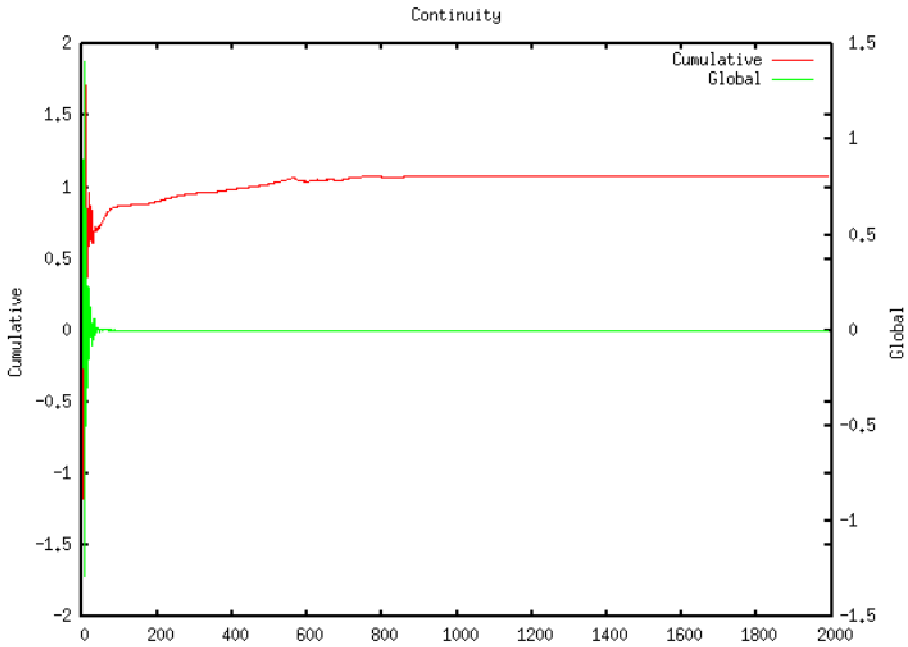
８. 計算結果の比較

No16　realizableKE,upwind,GAMG smoothSolver



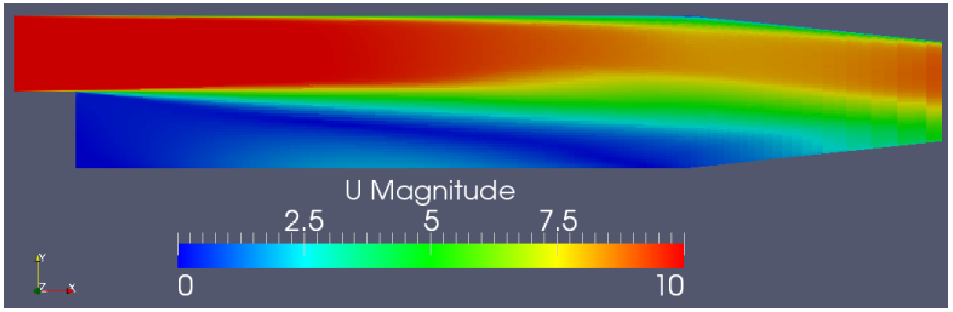


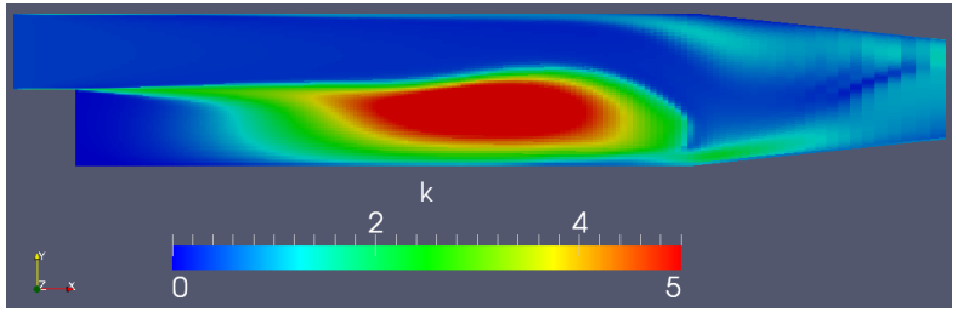


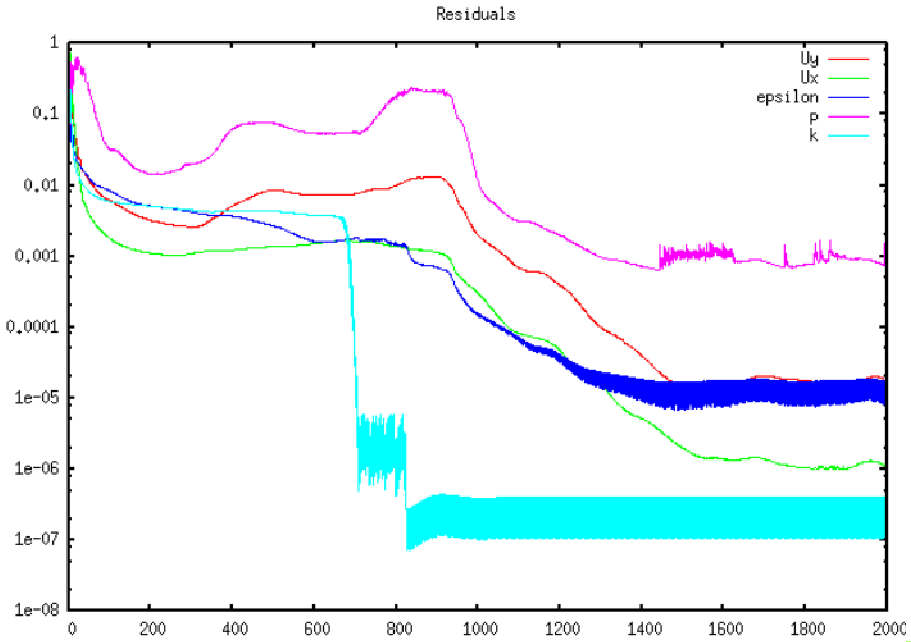


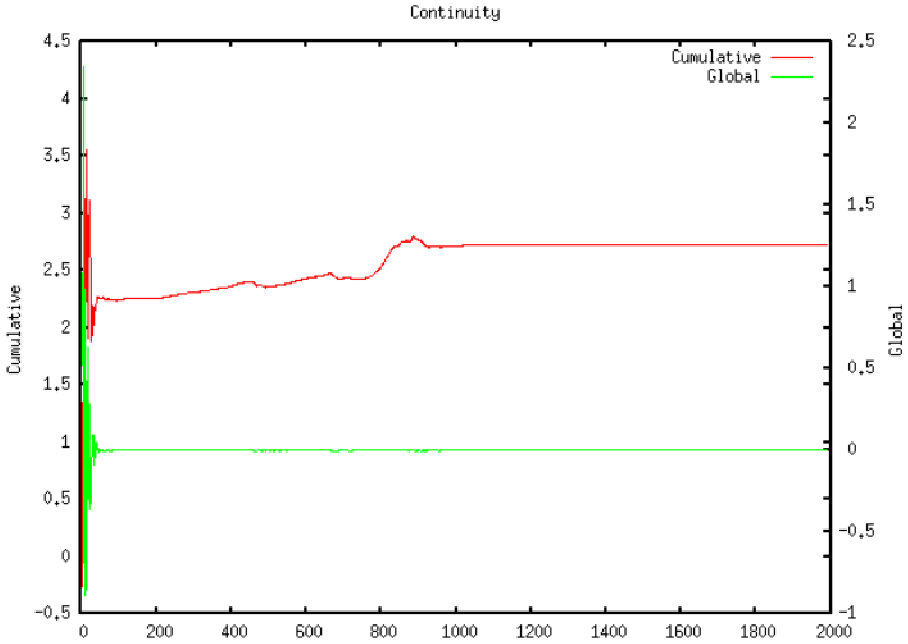
８. 計算結果の比較

No17　realizableKE,linear,GAMG smoothSolver



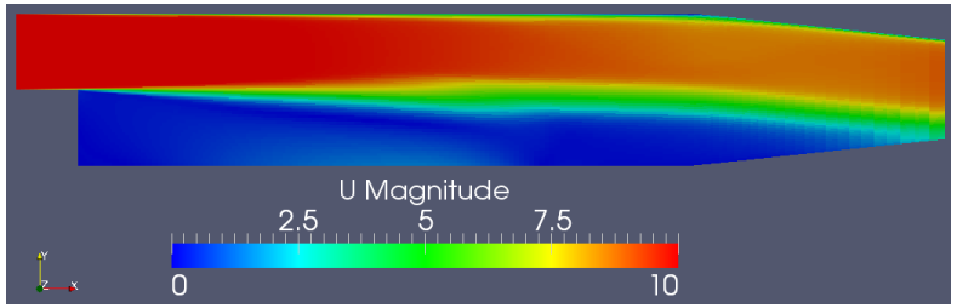


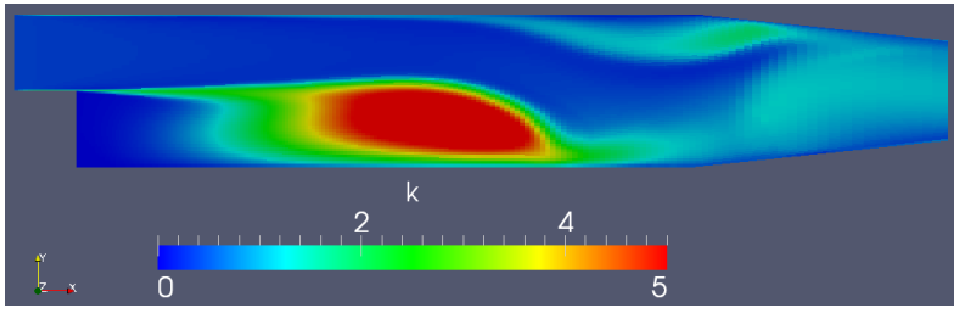


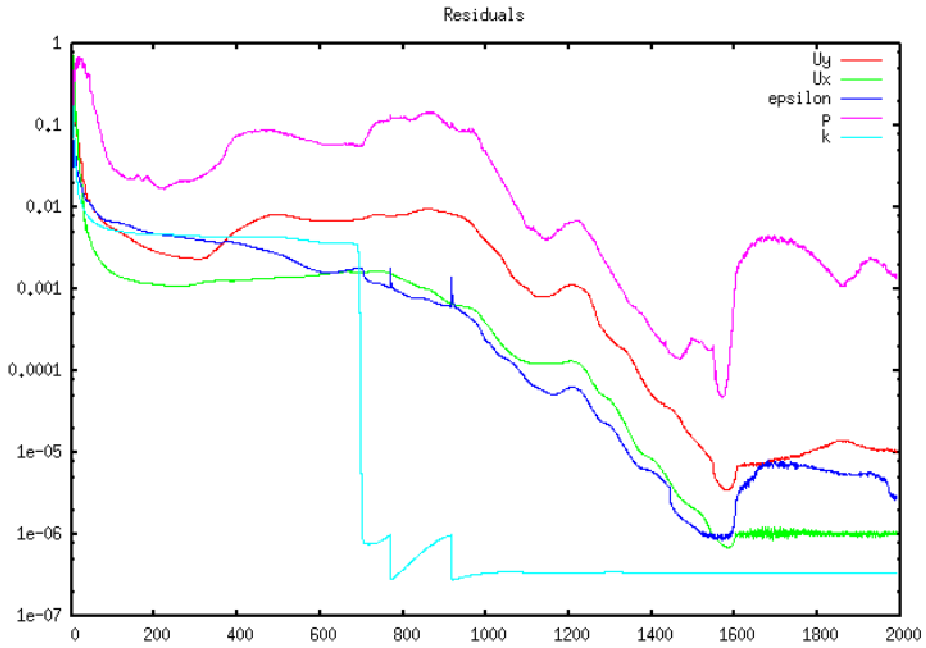


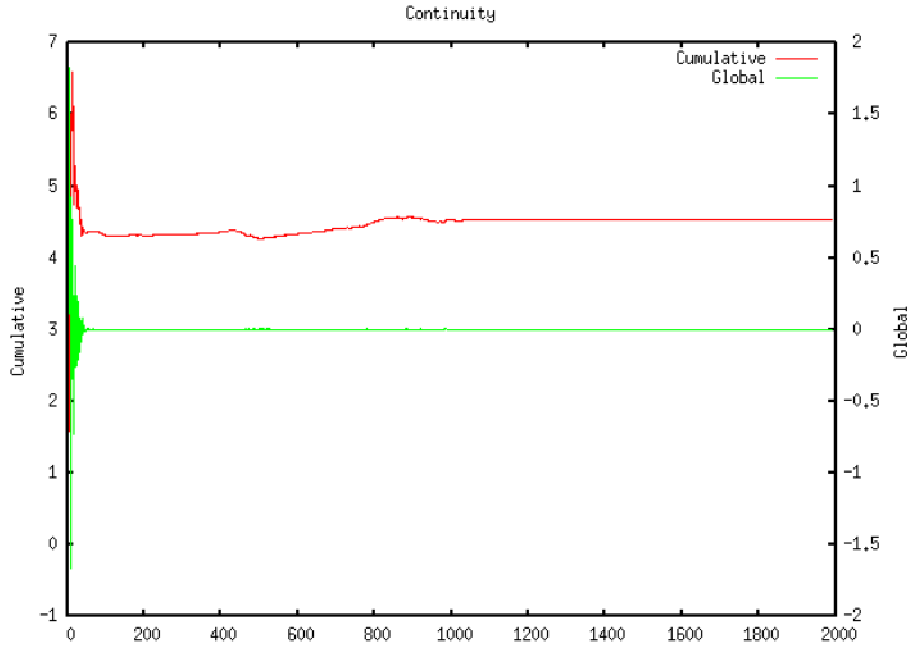
８. 計算結果の比較

No18　realizableKE,TVD,GAMG smoothSolver



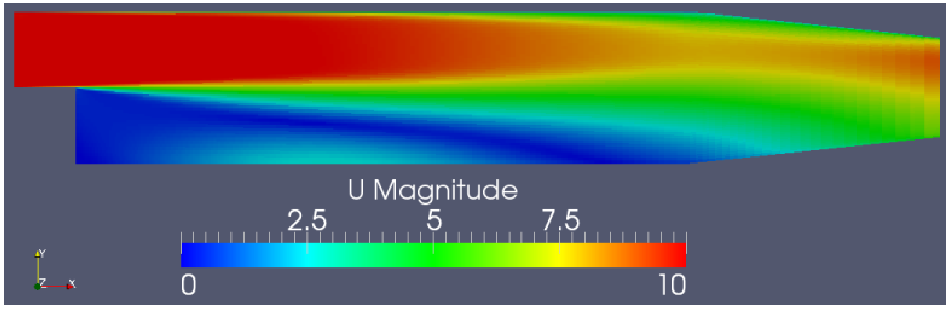


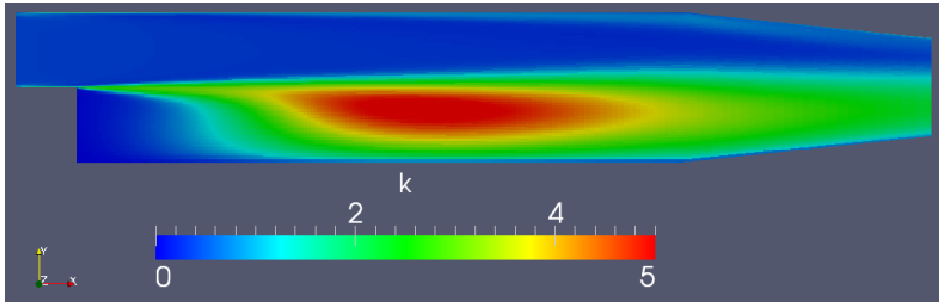


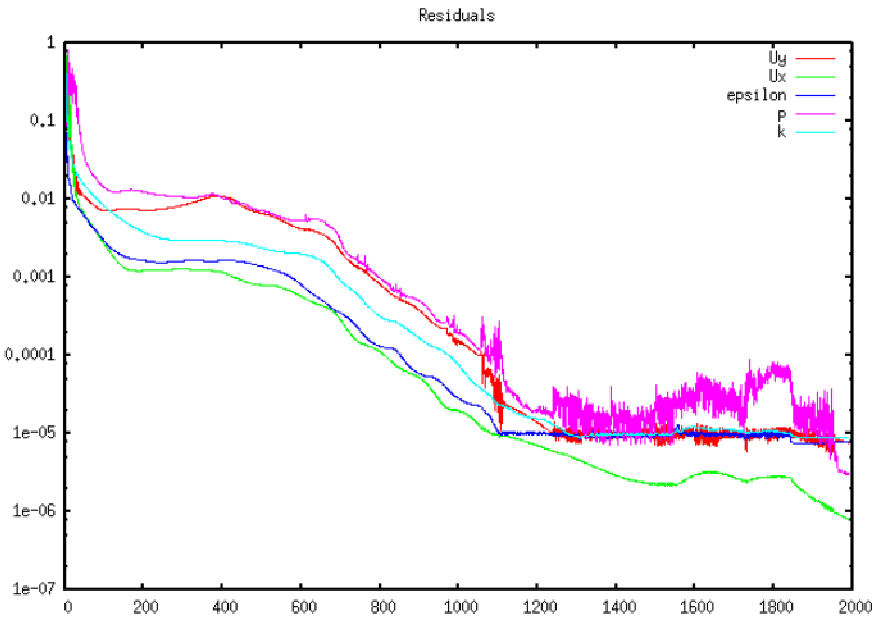


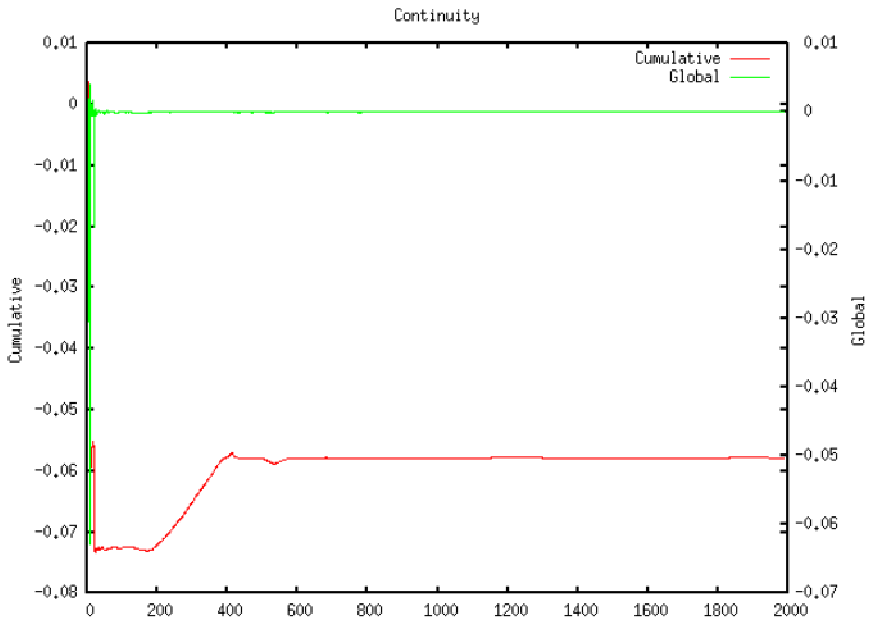
８. 計算結果の比較

No19　kEpsilon,upwind,PCG PBiCG





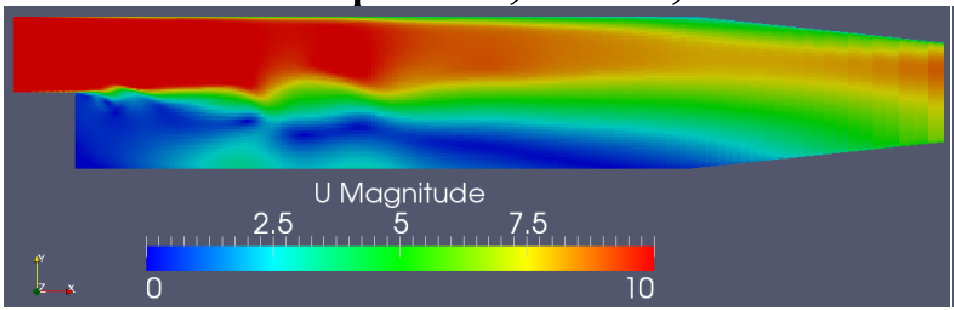


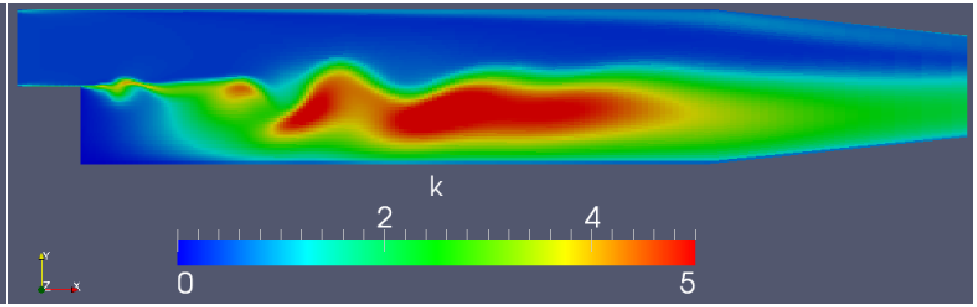


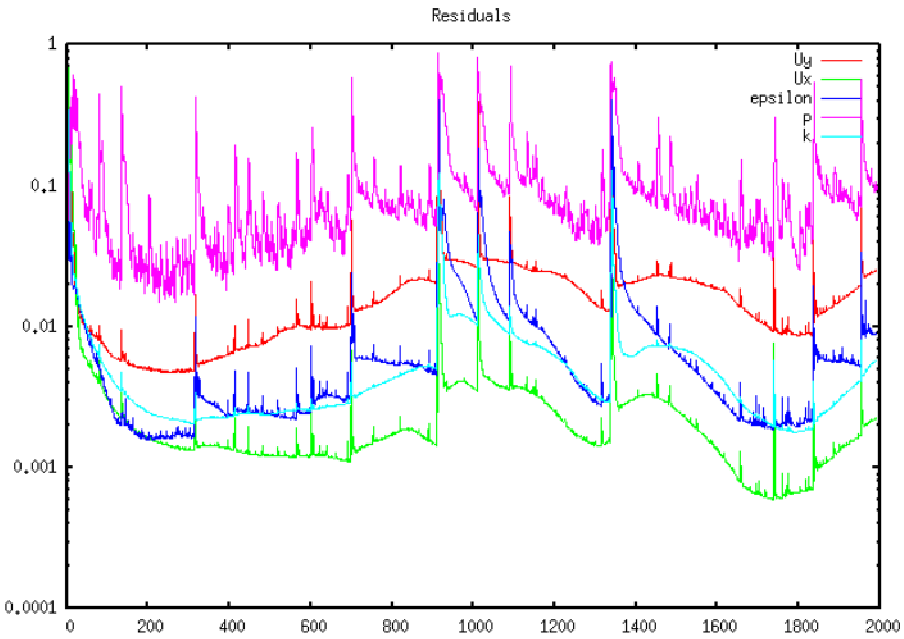
８．計算結果の比較

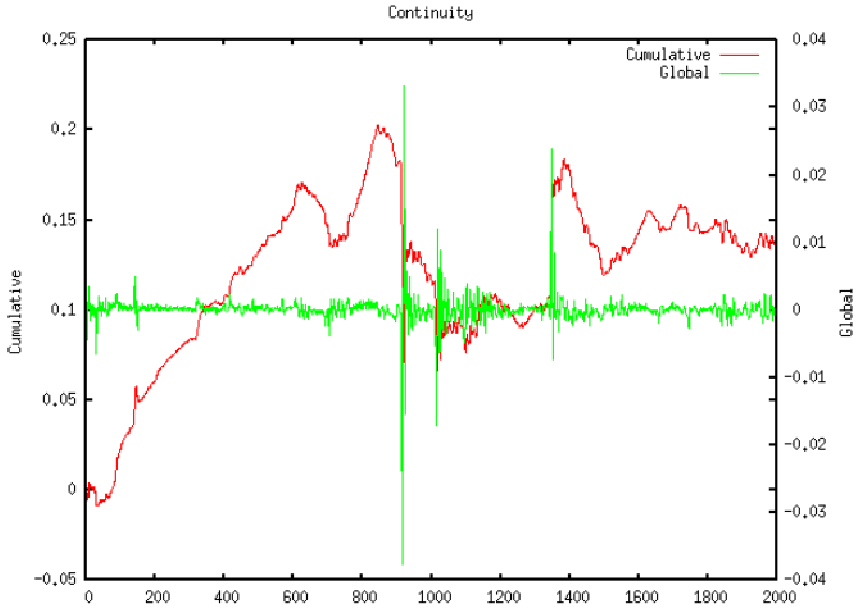
No20　kEpsilon,linear,PCG PBiCG

収束が安定しない



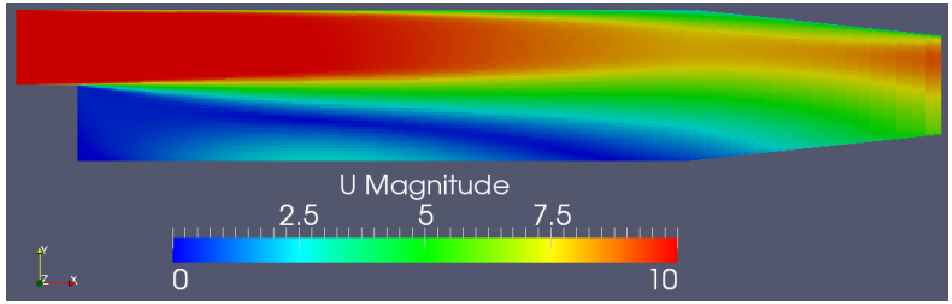


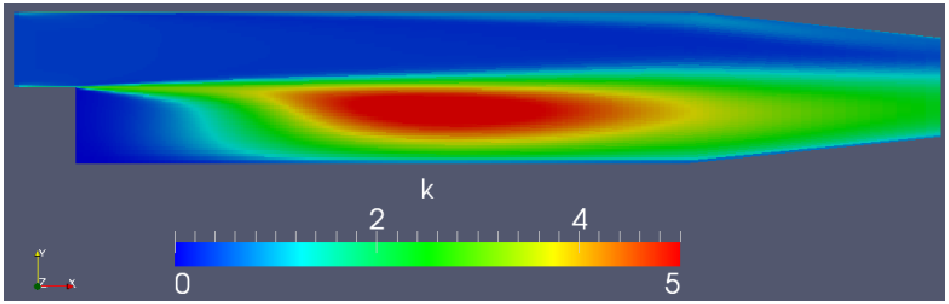


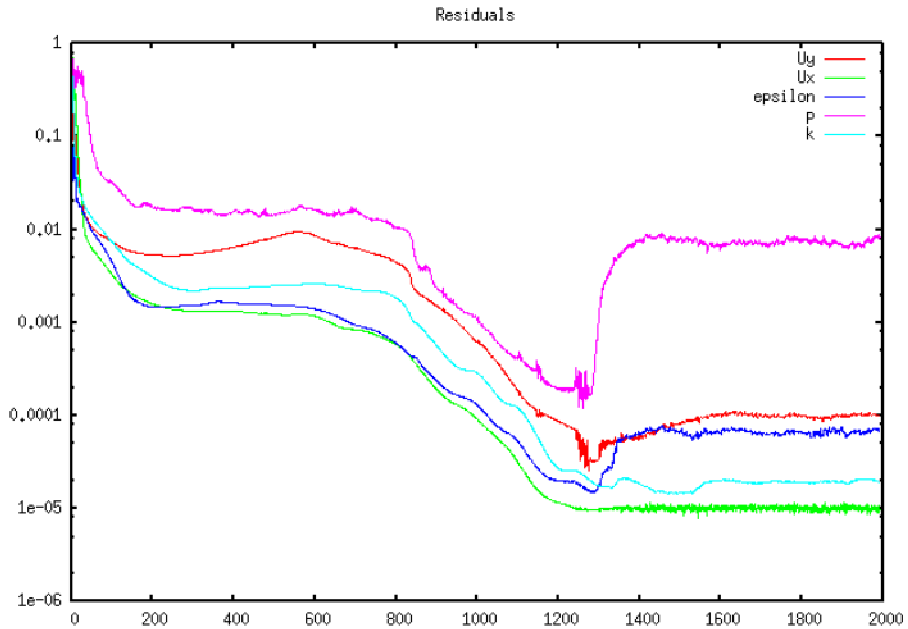


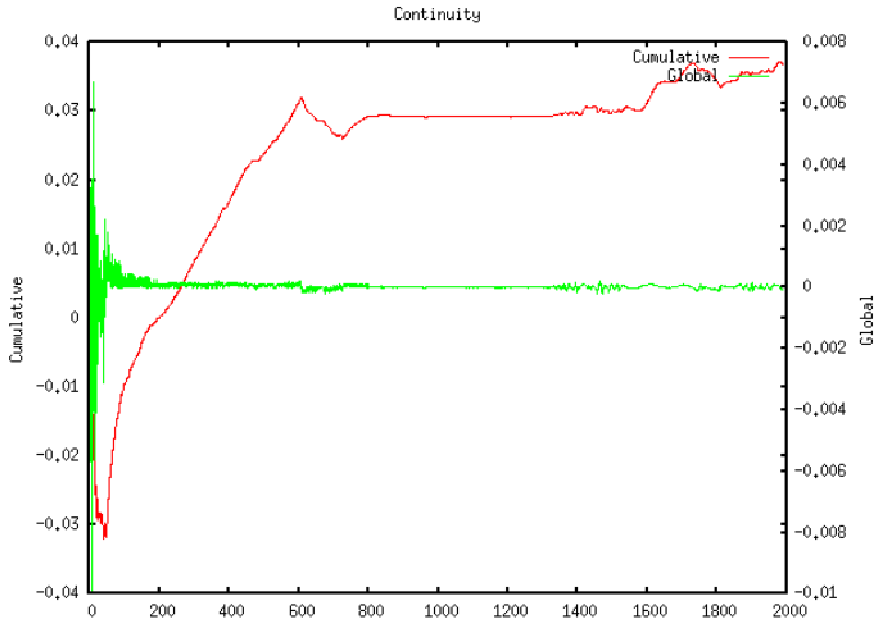
８．計算結果の比較

No21　kEpsilon,TVD,PCG PBiCG



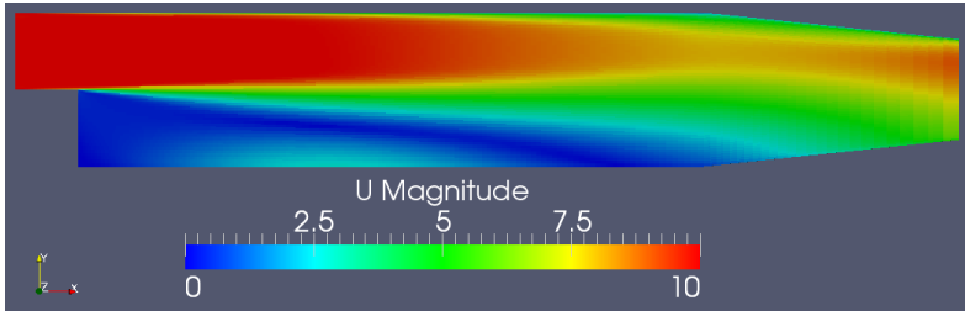


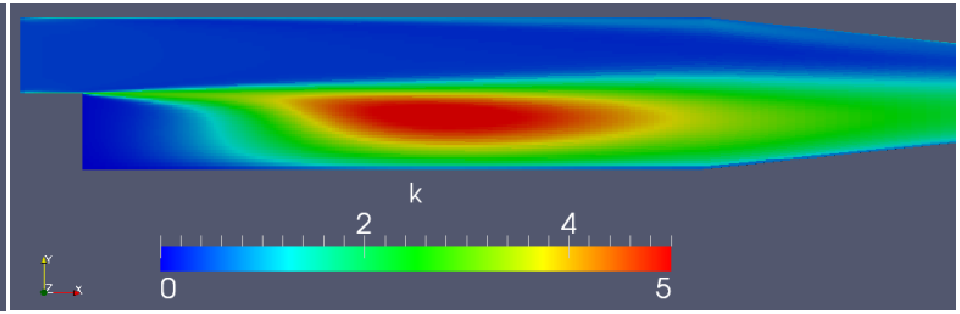


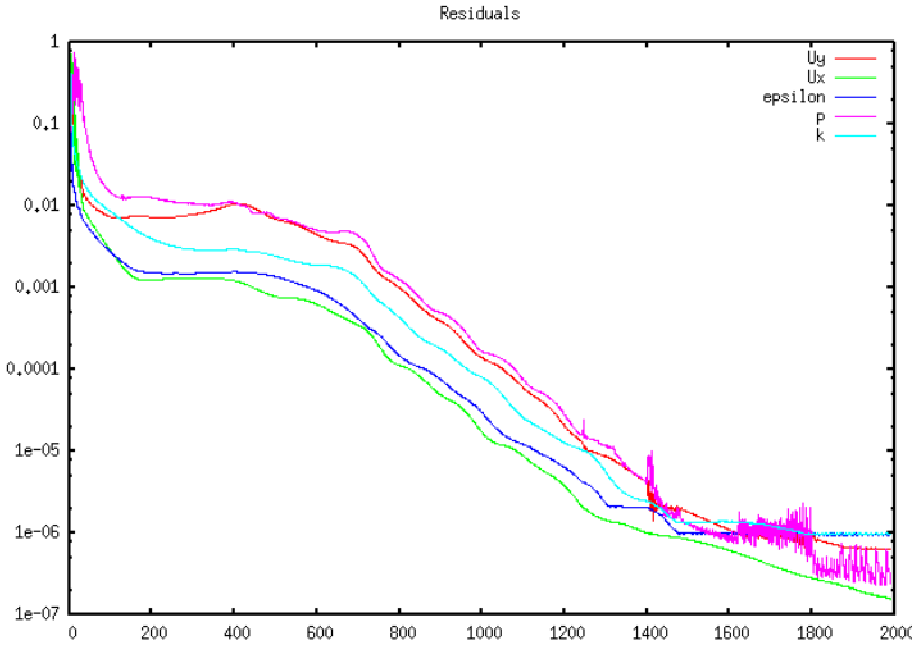


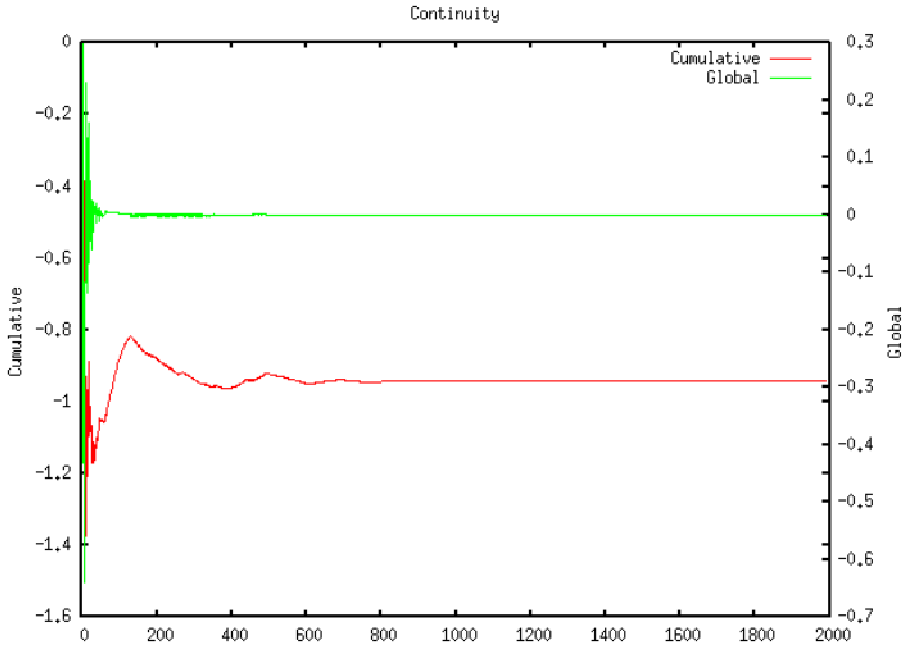
８．計算結果の比較

No22　kEpsilon,upwind,GAMG smoothSolver









８. 計算結果の比較

No23　kEpsilon,linear, GAMG smoothSolver

```
#0  Foam::error::printStack(Foam::Ostream&) in "/home/sakuramaru/OpenFOAM/OpenFOAM-
1.7.x/lib/linux64GccDPOpt/libOpenFOAM.so"
#1  Foam::sigFpe::sigFpeHandler(int) in "/home/sakuramaru/OpenFOAM/OpenFOAM-
1.7.x/lib/linux64GccDPOpt/libOpenFOAM.so"
#2   in "/lib/libc.so.6"
#3  Foam::PCG::solve(Foam::Field<double>&, Foam::Field<double> const&, unsigned char) const in
"/home/sakuramaru/OpenFOAM/OpenFOAM-1.7.x/lib/linux64GccDPOpt/libOpenFOAM.so"
#4  Foam::GAMGSolver::solveCoarsestLevel(Foam::Field<double>&, Foam::Field<double> const&) const in
"/home/sakuramaru/OpenFOAM/OpenFOAM-1.7.x/lib/linux64GccDPOpt/libOpenFOAM.so"
#5  Foam::GAMGSolver::Vcycle(Foam::PtrList<Foam::lduMatrix::smoother> const&, Foam::Field<double>&,
Foam::Field<double> const&, Foam::Field<double>&, Foam::Field<double>&, Foam::Field<double>&,
Foam::PtrList<Foam::Field<double> >&, Foam::PtrList<Foam::Field<double> >&, unsigned char) const in
"/home/sakuramaru/OpenFOAM/OpenFOAM-1.7.x/lib/linux64GccDPOpt/libOpenFOAM.so"
#6  Foam::GAMGSolver::solve(Foam::Field<double>&, Foam::Field<double> const&, unsigned char) const in
"/home/sakuramaru/OpenFOAM/OpenFOAM-1.7.x/lib/linux64GccDPOpt/libOpenFOAM.so"
#7  Foam::fvMatrix<double>::solve(Foam::dictionary const&) in "/home/sakuramaru/OpenFOAM/OpenFOAM-
1.7.x/lib/linux64GccDPOpt/libfiniteVolume.so"
#8
 in "/home/sakuramaru/OpenFOAM/OpenFOAM-1.7.x/applications/bin/linux64GccDPOpt/simpleFoam"
#9  __libc_start_main in "/lib/libc.so.6"
#10
 in "/home/sakuramaru/OpenFOAM/OpenFOAM-1.7.x/applications/bin/linux64GccDPOpt/simpleFoam"
Floating point exception
```

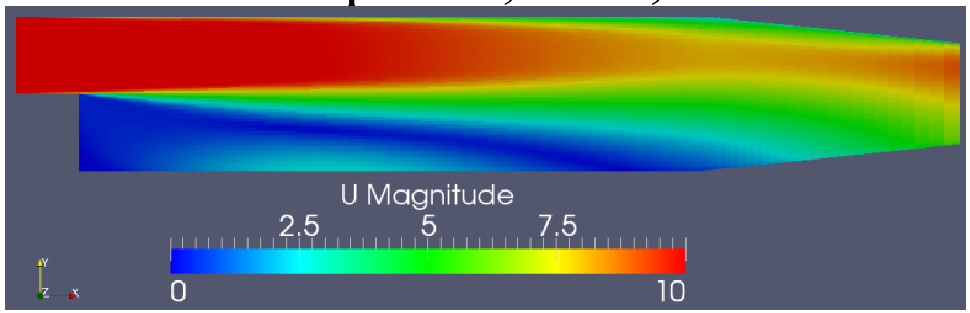
８．計算結果の比較

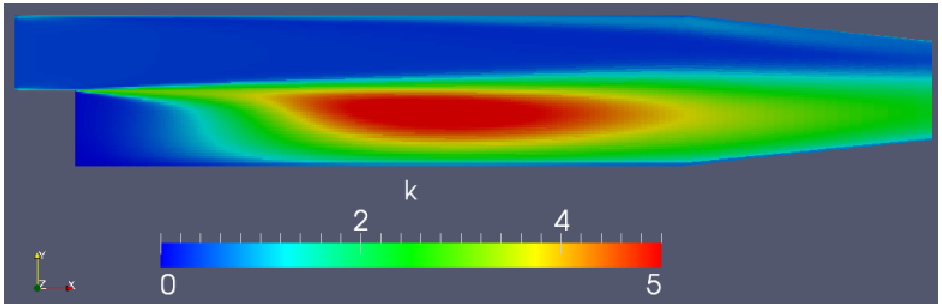
No23　kEpsilon,linear, GAMG smoothSolver

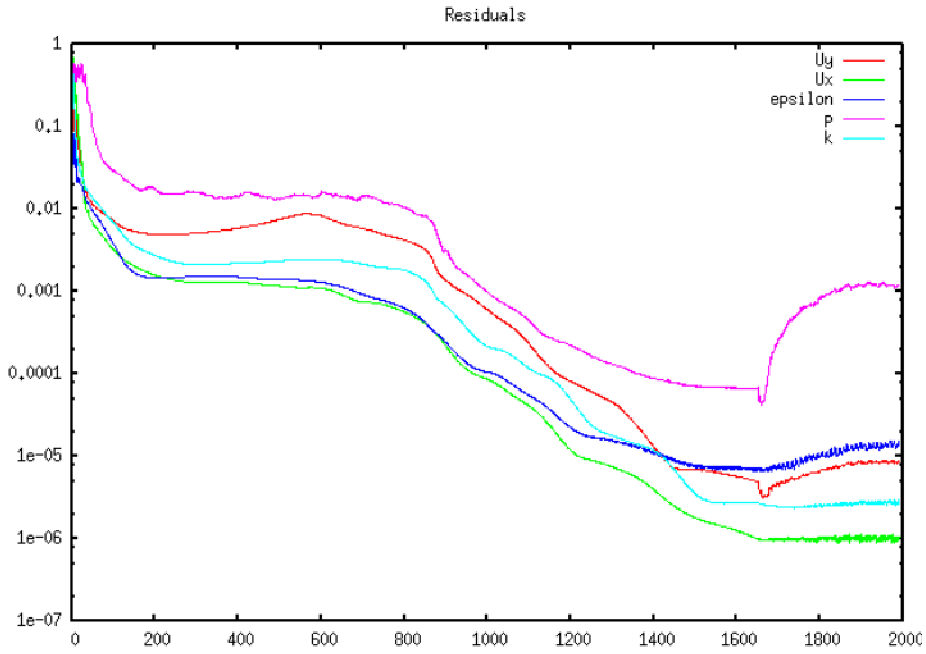
Time = 1030
smoothSolver: Solving for Ux, Initial residual = 0.764566, Final residual = 0.0397834, No Iterations 3
smoothSolver: Solving for Uy, Initial residual = 0.209879, Final residual = 0.0148312, No Iterations 4
GAMG: Solving for p, Initial residual = 1, Final residual = 0.0392114, No Iterations 3
time step continuity errors : sum local = 1.31161e+43, global = -1.22957e+42, cumulative = -1.22957e+42
smoothSolver: Solving for epsilon, Initial residual = 1, Final residual = 0.0862734, No Iterations 4
bounding epsilon, min: -3.38166e+70 max: 1.92608e+73 average: 1.61417e+69
smoothSolver: Solving for k, Initial residual = 1, Final residual = 0.041405, No Iterations 3
bounding k, min: -6.30268e+57 max: 5.32925e+58 average: 6.00644e+55
ExecutionTime = 40.78 s ClockTime = 40 s

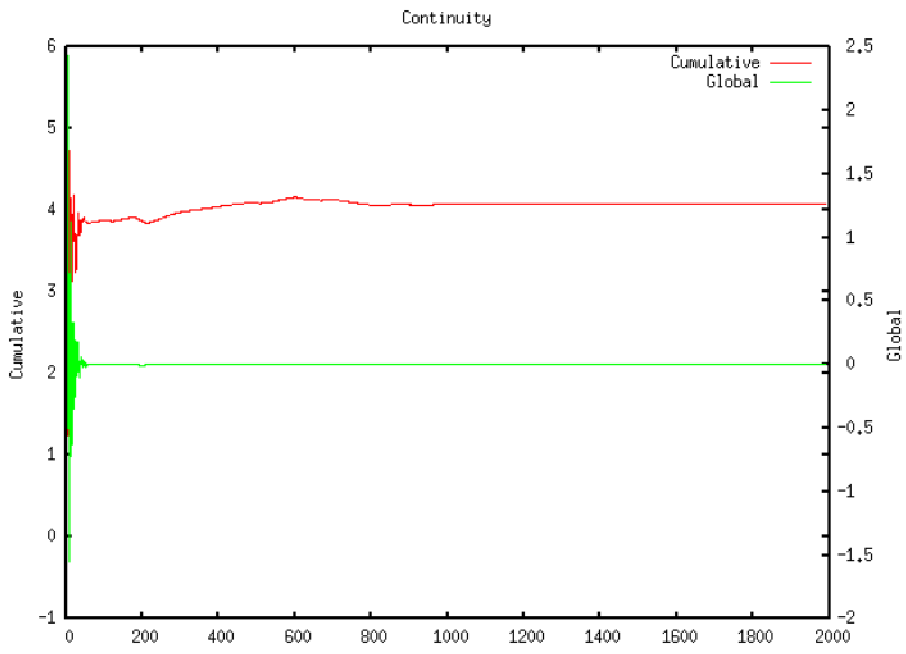
８. 計算結果の比較

No24　kEpsilon,TVD,GAMG smoothSolver





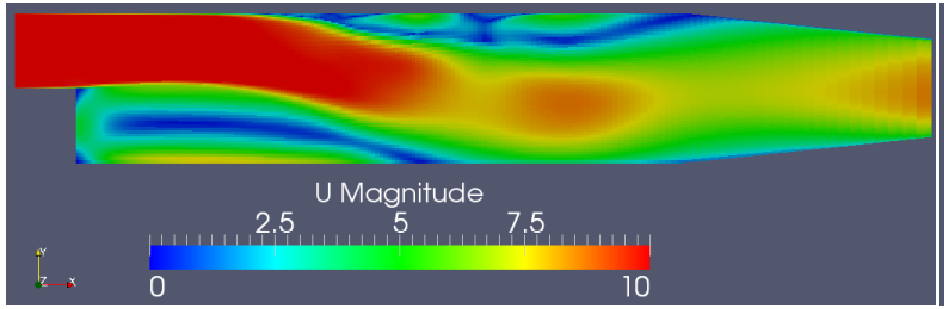


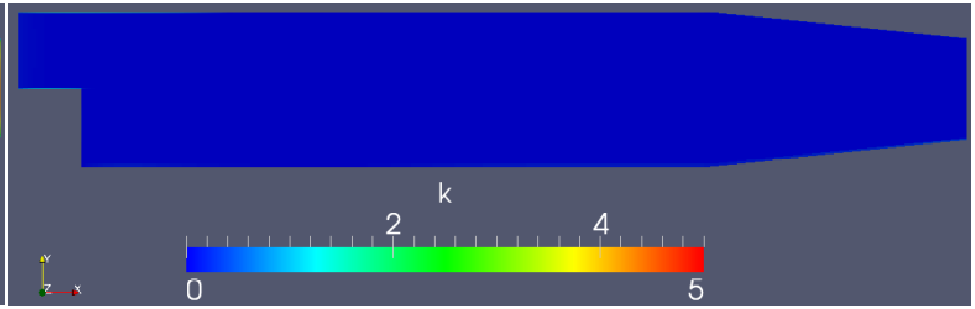


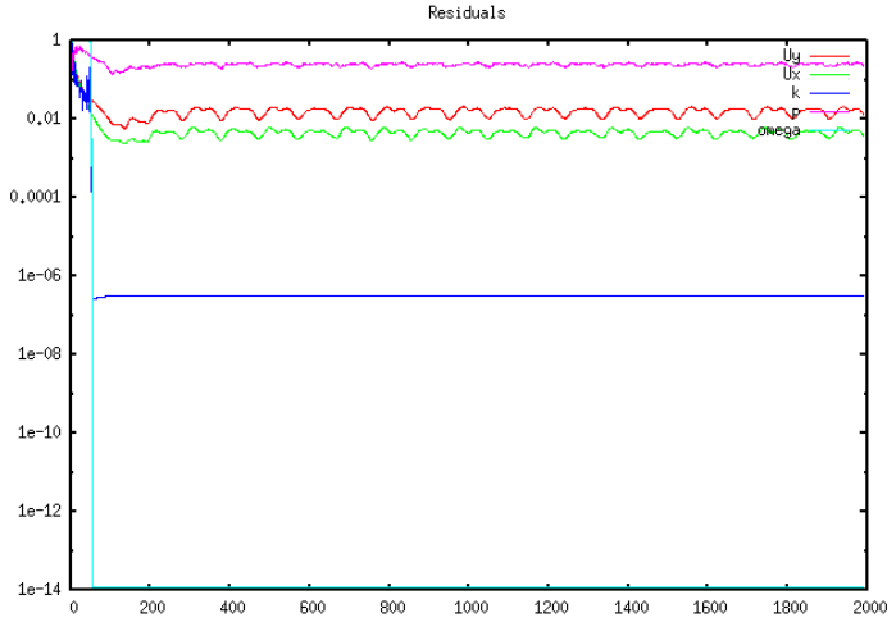
８．計算結果の比較

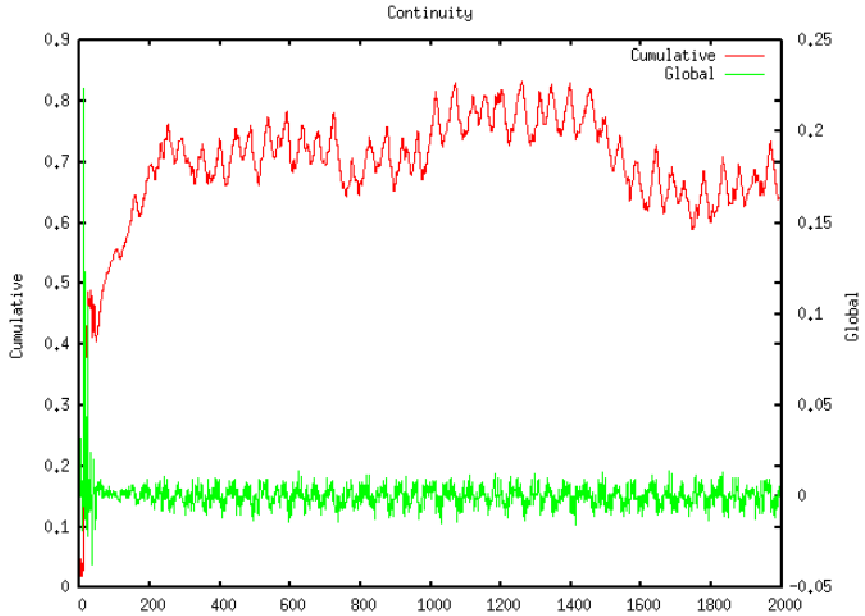
No25　kOmegaSST,upwind,PCG PBiCG

収束がおかしい









８．計算結果の比較

No26　kOmgaSST,linear, PCG PBiCG

```
#0  Foam::error::printStack(Foam::Ostream&) in "/home/sakuramaru/OpenFOAM/OpenFOAM-
1.7.x/lib/linux64GccDPOpt/libOpenFOAM.so"
#1  Foam::sigFpe::sigFpeHandler(int) in "/home/sakuramaru/OpenFOAM/OpenFOAM-
1.7.x/lib/linux64GccDPOpt/libOpenFOAM.so"
#2   in "/lib/libc.so.6"
#3  Foam::PBiCG::solve(Foam::Field<double>&, Foam::Field<double> const&, unsigned char) const in
"/home/sakuramaru/OpenFOAM/OpenFOAM-1.7.x/lib/linux64GccDPOpt/libOpenFOAM.so"
#4  Foam::fvMatrix<double>::solve(Foam::dictionary const&) in "/home/sakuramaru/OpenFOAM/OpenFOAM-
1.7.x/lib/linux64GccDPOpt/libfiniteVolume.so"
#5  Foam::lduMatrix::solverPerformance Foam::solve<double>(Foam::tmp<Foam::fvMatrix<double> > const&) in
"/home/sakuramaru/OpenFOAM/OpenFOAM-1.7.x/lib/linux64GccDPOpt/libincompressibleRASModels.so"
#6  Foam::incompressible::RASModels::kOmegaSST::correct() in "/home/sakuramaru/OpenFOAM/OpenFOAM-
1.7.x/lib/linux64GccDPOpt/libincompressibleRASModels.so"
#7
 in "/home/sakuramaru/OpenFOAM/OpenFOAM-1.7.x/applications/bin/linux64GccDPOpt/simpleFoam"
#8  __libc_start_main in "/lib/libc.so.6"
#9
 in "/home/sakuramaru/OpenFOAM/OpenFOAM-1.7.x/applications/bin/linux64GccDPOpt/simpleFoam"
Floating point exception
```

８. 計算結果の比較

No26　kOmgaSST,linear, PCG PBiCG

Time = 13
DILUPBiCG: Solving for Ux, Initial residual = 0.0882911, Final residual = 0.00176956, No Iterations 1
DILUPBiCG: Solving for Uy, Initial residual = 0.0589643, Final residual = 9.40842e-05, No Iterations 2
DICPCG: Solving for p, Initial residual = 0.293575, Final residual = 0.00284045, No Iterations 141
time step continuity errors : sum local = 1.02058, global = 0.0136782, cumulative = 0.236392
DILUPBiCG: Solving for omega, Initial residual = 1, Final residual = 4.15499e-18, No Iterations 1
bounding omega, min: -8.83428e+108 max: 2.08912e+109 average: 1.05743e+105
DILUPBiCG: Solving for k, Initial residual = 0.00493906, Final residual = 1.18318e-22, No Iterations 1
bounding k, min: -0.218238 max: 3.12828 average: 0.0177913
ExecutionTime = 2.25 s ClockTime = 2 s

８．計算結果の比較

No27　kOmgaSST,TVD, PCG PBiCG

```
#0  Foam::error::printStack(Foam::Ostream&) in "/home/sakuramaru/OpenFOAM/OpenFOAM-
1.7.x/lib/linux64GccDPOpt/libOpenFOAM.so"
#1  Foam::sigFpe::sigFpeHandler(int) in "/home/sakuramaru/OpenFOAM/OpenFOAM-
1.7.x/lib/linux64GccDPOpt/libOpenFOAM.so"
#2   in "/lib/libc.so.6"
#3  Foam::LimitedScheme<double, Foam::limitedLinearLimiter<Foam::NVDTVD>,
Foam::limitFuncs::magSqr>::limiter(Foam::GeometricField<double, Foam::fvPatchField, Foam::volMesh> const&) const in
"/home/sakuramaru/OpenFOAM/OpenFOAM-1.7.x/lib/linux64GccDPOpt/libfiniteVolume.so"
#4  Foam::limitedSurfaceInterpolationScheme<double>::weights(Foam::GeometricField<double, Foam::fvPatchField,
Foam::volMesh> const&) const in "/home/sakuramaru/OpenFOAM/OpenFOAM-1.7.x/lib/linux64GccDPOpt/libfiniteVolume.so"
#5  Foam::fv::gaussConvectionScheme<double>::fvmDiv(Foam::GeometricField<double, Foam::fvsPatchField,
Foam::surfaceMesh> const&, Foam::GeometricField<double, Foam::fvPatchField, Foam::volMesh>&) const in
"/home/sakuramaru/OpenFOAM/OpenFOAM-1.7.x/lib/linux64GccDPOpt/libfiniteVolume.so"
#6  Foam::tmp<Foam::fvMatrix<double> > Foam::fvm::div<double>(Foam::GeometricField<double, Foam::fvsPatchField,
Foam::surfaceMesh> const&, Foam::GeometricField<double, Foam::fvPatchField, Foam::volMesh>&, Foam::word const&) in
"/home/sakuramaru/OpenFOAM/OpenFOAM-1.7.x/lib/linux64GccDPOpt/libincompressibleRASModels.so"
#7  Foam::tmp<Foam::fvMatrix<double> > Foam::fvm::div<double>(Foam::GeometricField<double, Foam::fvsPatchField,
Foam::surfaceMesh> const&, Foam::GeometricField<double, Foam::fvPatchField, Foam::volMesh>&) in
"/home/sakuramaru/OpenFOAM/OpenFOAM-1.7.x/lib/linux64GccDPOpt/libincompressibleRASModels.so"
#8  Foam::incompressible::RASModels::kOmegaSST::correct() in "/home/sakuramaru/OpenFOAM/OpenFOAM-
1.7.x/lib/linux64GccDPOpt/libincompressibleRASModels.so"
#9
 in "/home/sakuramaru/OpenFOAM/OpenFOAM-1.7.x/applications/bin/linux64GccDPOpt/simpleFoam"
#10  __libc_start_main in "/lib/libc.so.6"
#11
 in "/home/sakuramaru/OpenFOAM/OpenFOAM-1.7.x/applications/bin/linux64GccDPOpt/simpleFoam"
Floating point exception
```

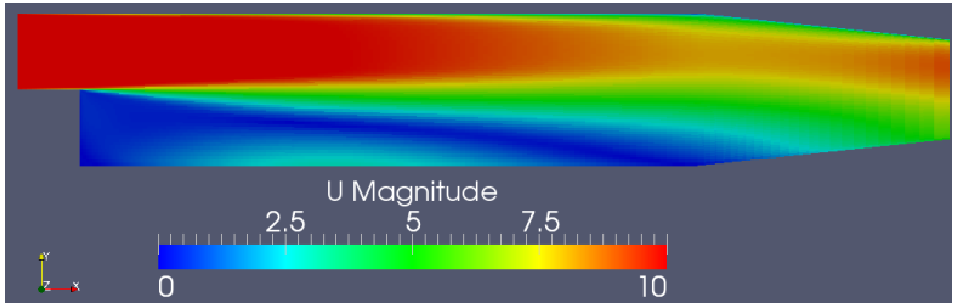
８. 計算結果の比較

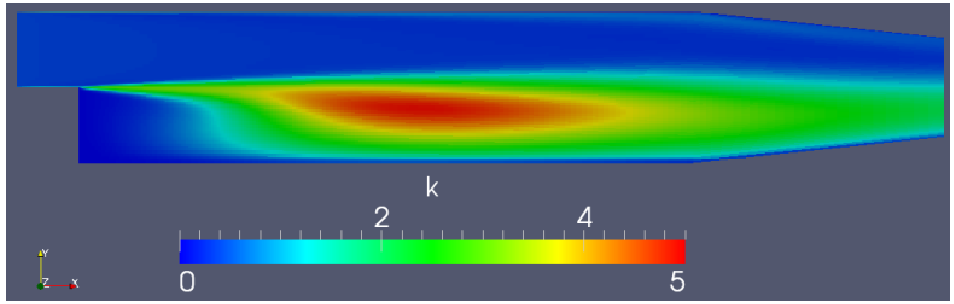
No27　kOmgaSST,TVD, PCG PBiCG

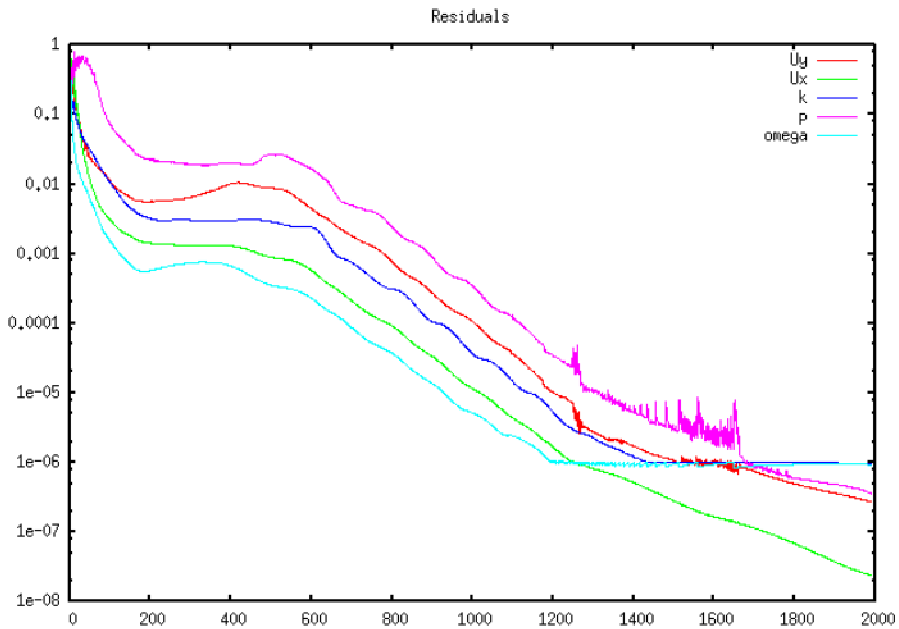
Time = 3
DILUPBiCG: Solving for Ux, Initial residual = 0.532708, Final residual = 0.033915, No Iterations 1
DILUPBiCG: Solving for Uy, Initial residual = 0.195258, Final residual = 0.0083128, No Iterations 1
DICPCG: Solving for p, Initial residual = 0.158018, Final residual = 0.00147366, No Iterations 141
time step continuity errors : sum local = 2.20071, global = 0.0175231, cumulative = 0.0482703
DILUPBiCG: Solving for omega, Initial residual = 1, Final residual = 0.0634549, No Iterations 26
bounding omega, min: -1.61642e+19 max: 1.62096e+21 average: 2.09599e+19
DILUPBiCG: Solving for k, Initial residual = 0.933999, Final residual = 7.4627e-17, No Iterations 1
ExecutionTime = 0.57 s ClockTime = 1 s

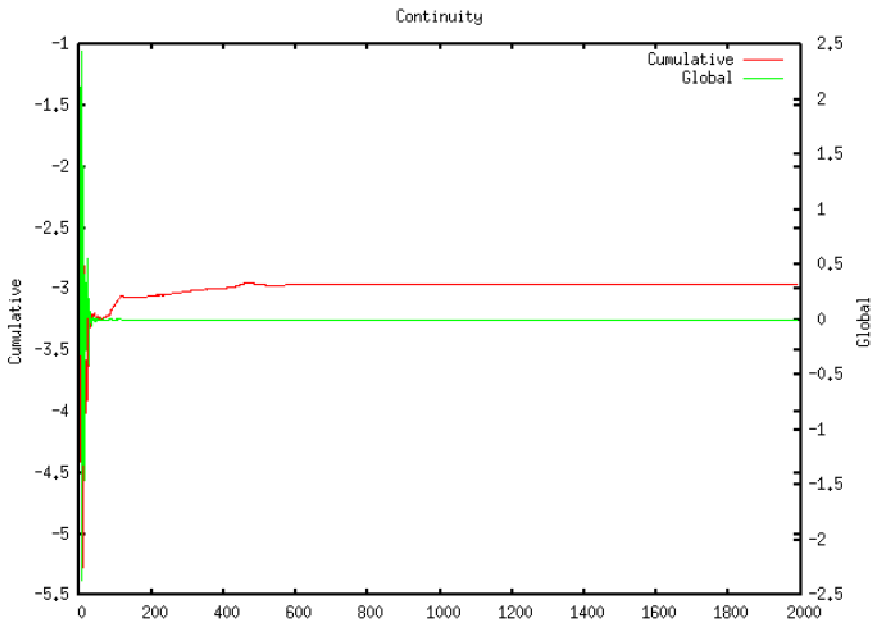
８. 計算結果の比較

No28　kOmegaSST,upwind,GAMG smoothSolver



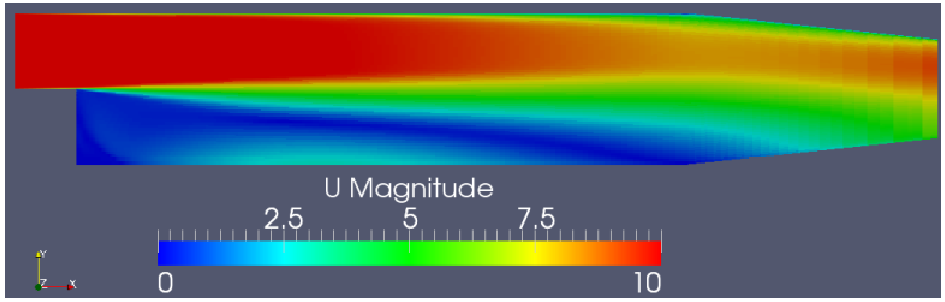


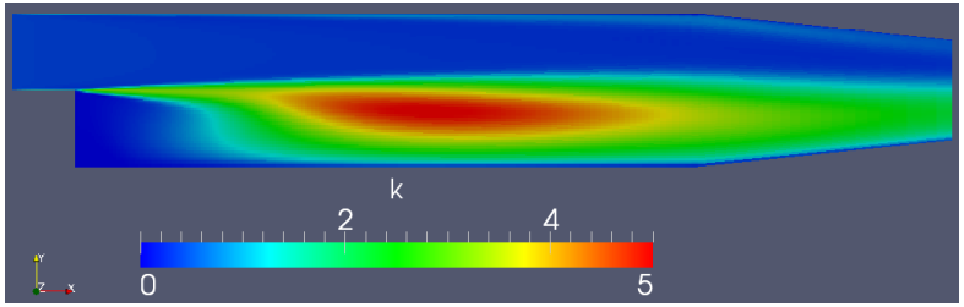


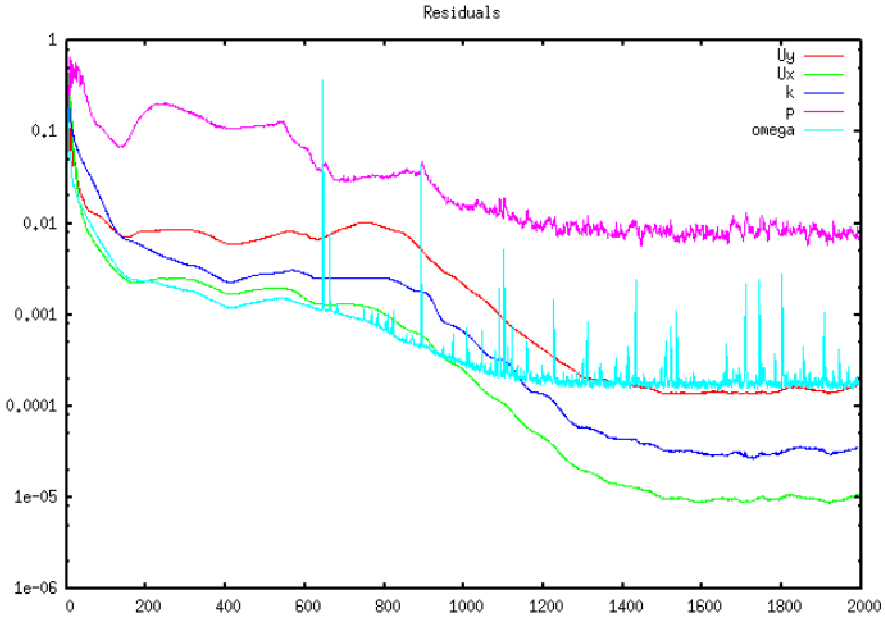


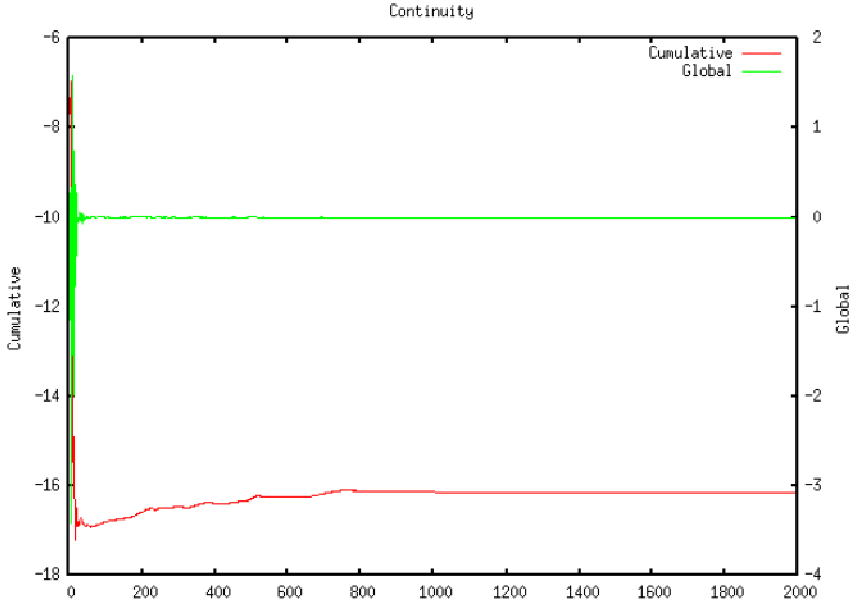
８. 計算結果の比較

No29　kOmegaSST,linear,GAMG smoothSolver



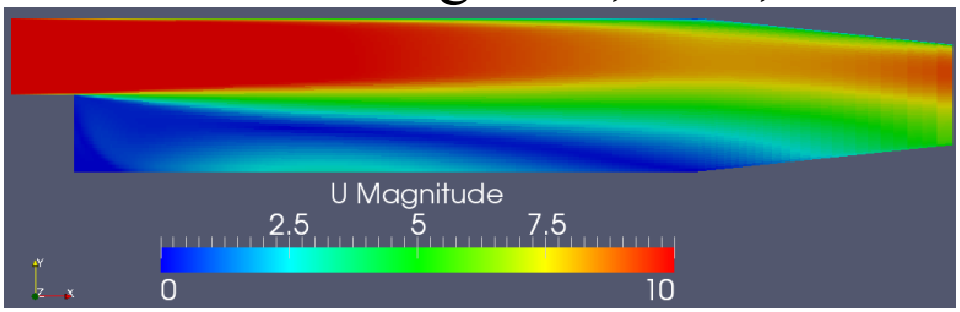


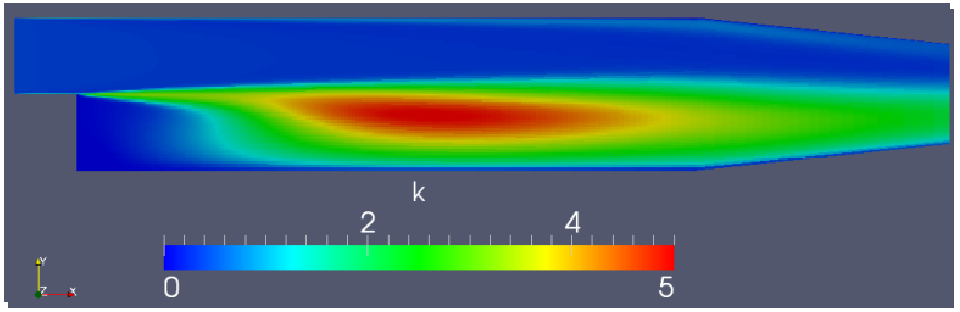


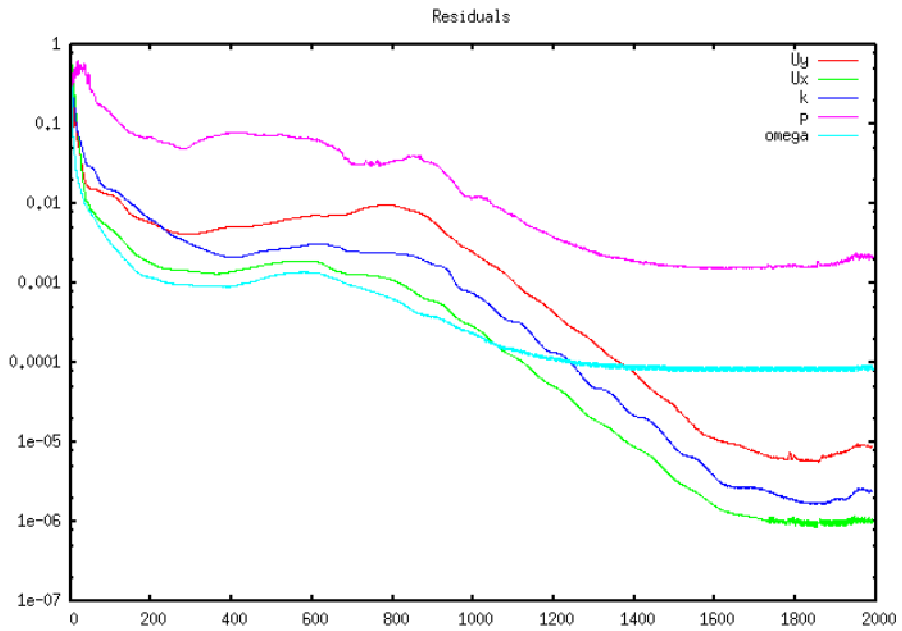


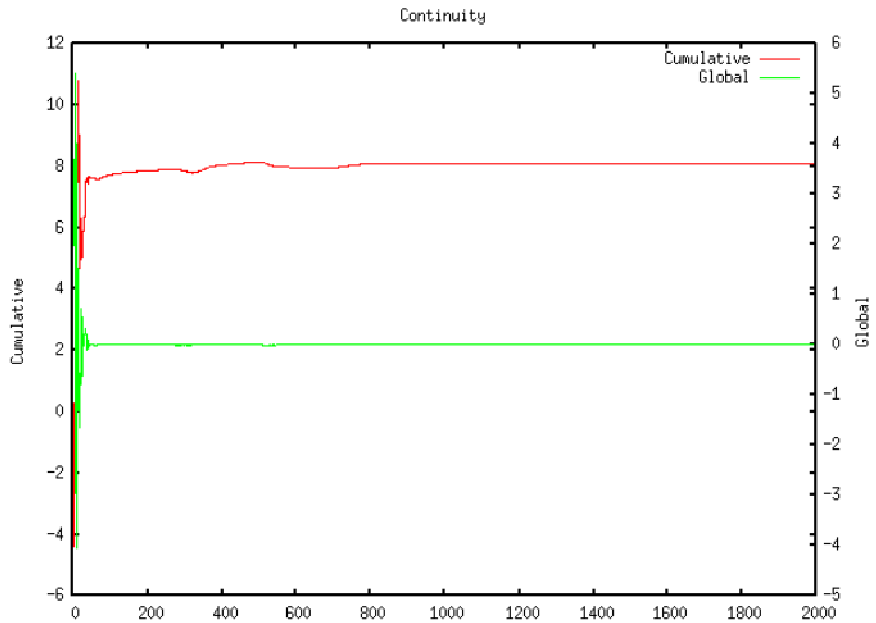
８. 計算結果の比較

No30　kOmegaSST,TVD,GAMG smoothSolver



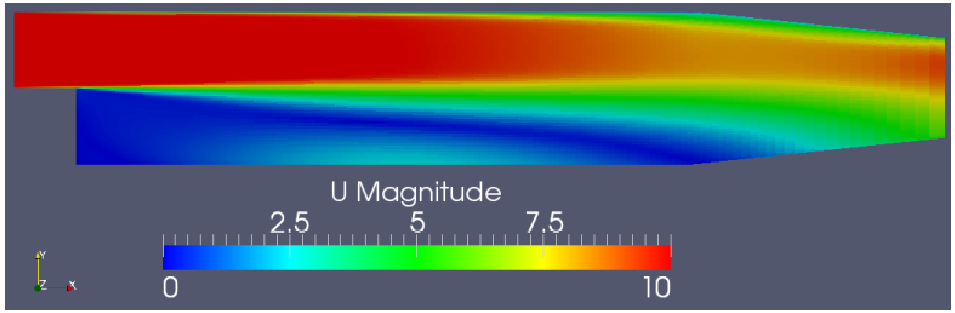


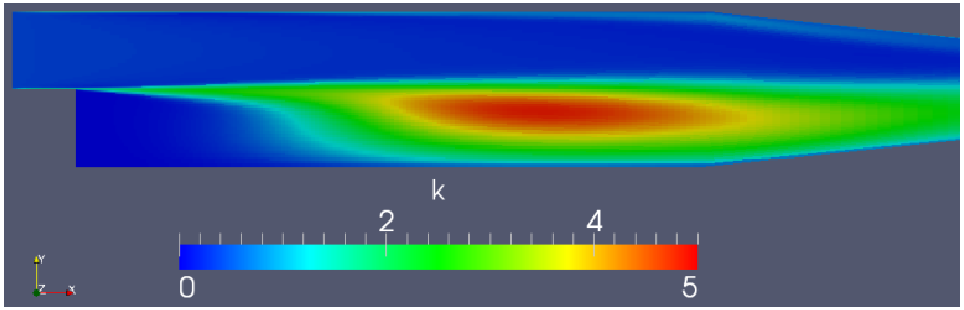


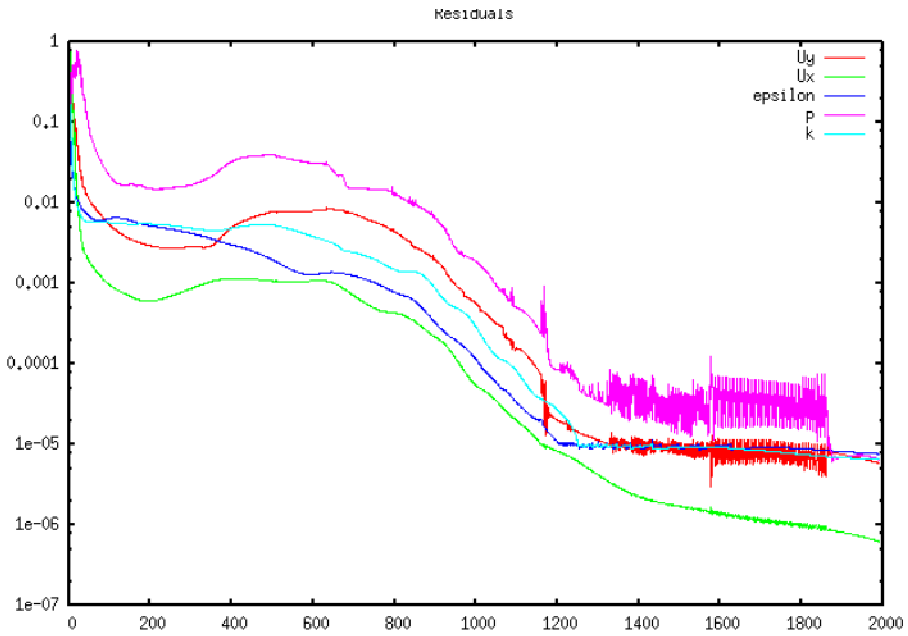


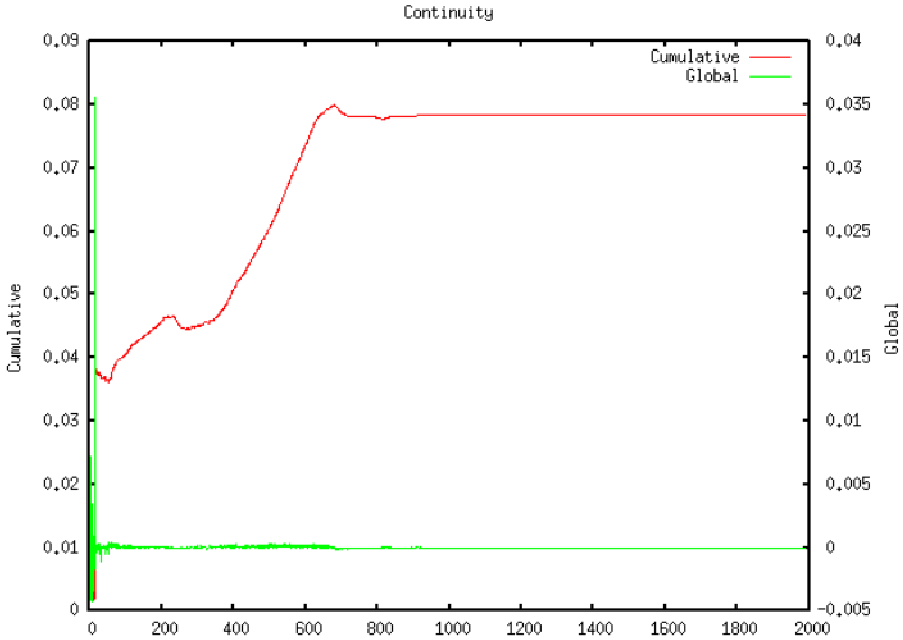
８. 計算結果の比較

No31　realizableKE,upwind,PCG PBiCG





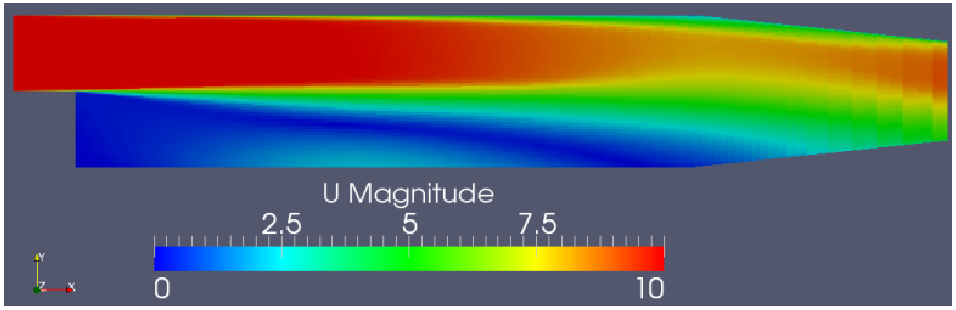


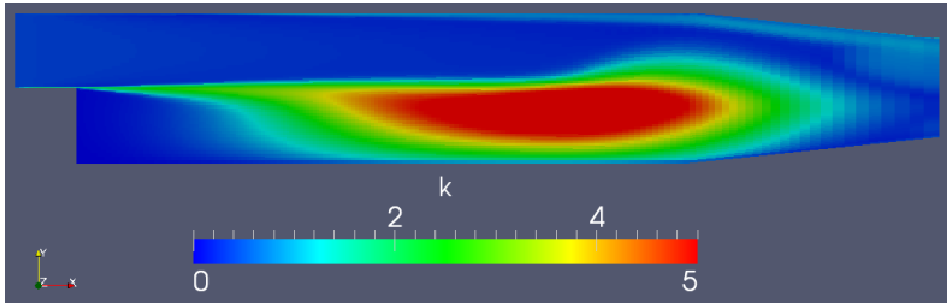


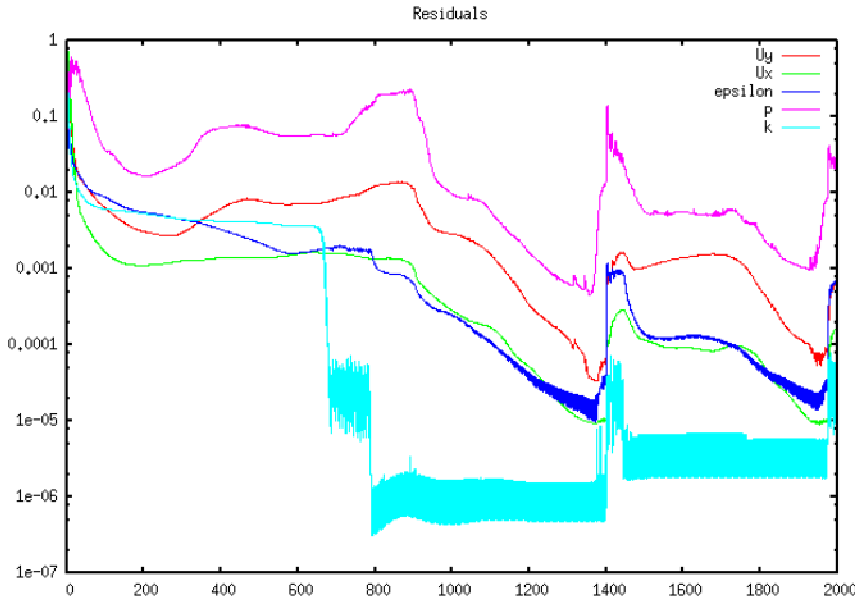
８．計算結果の比較

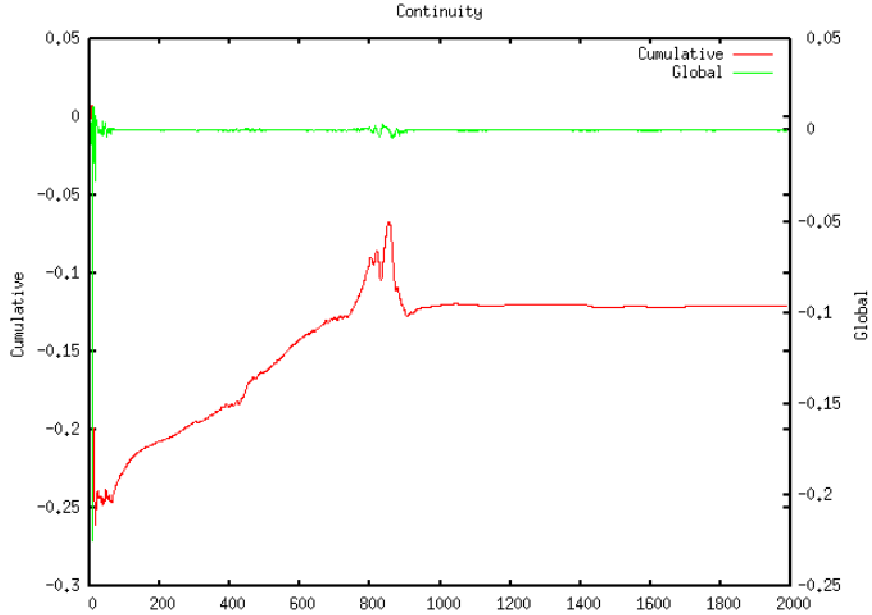
No32　realizableKE,linear,PCG PBiCG

収束が安定しない



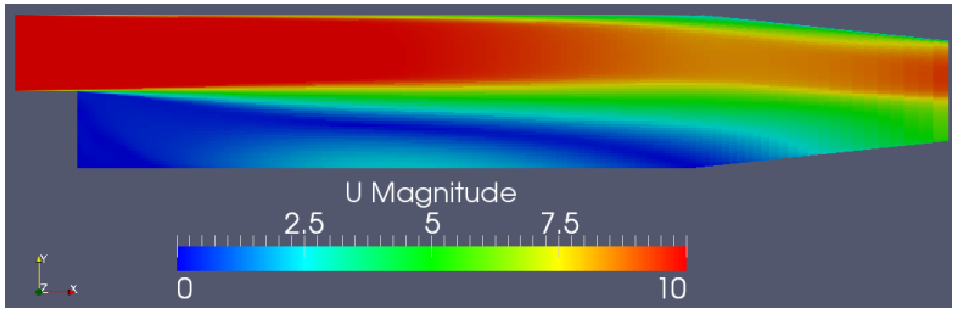


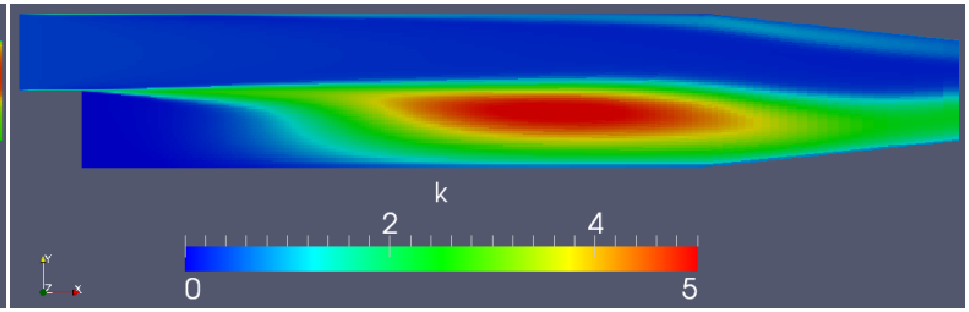


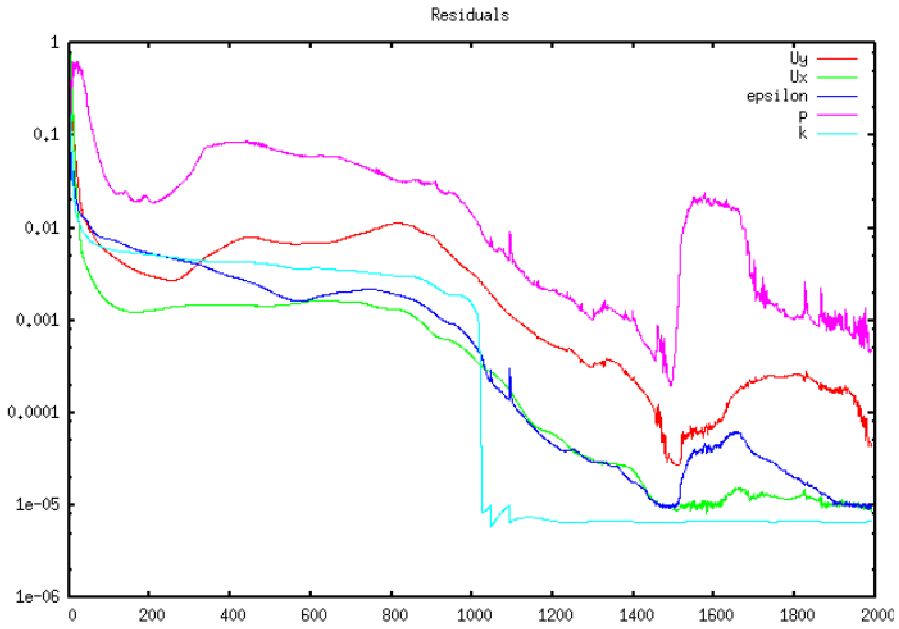


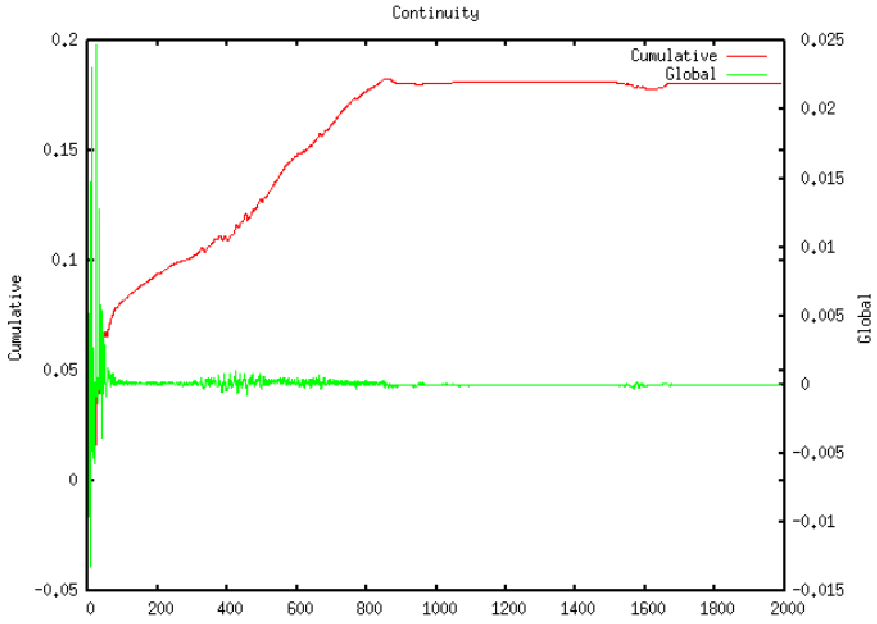
８．計算結果の比較

No33　realizableKE,TVD,PCG PBiCG



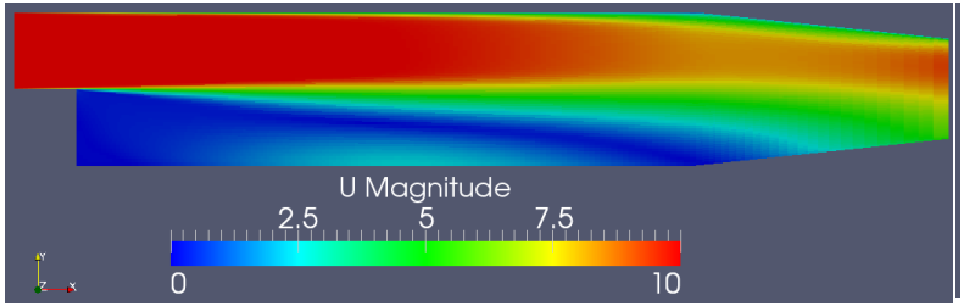


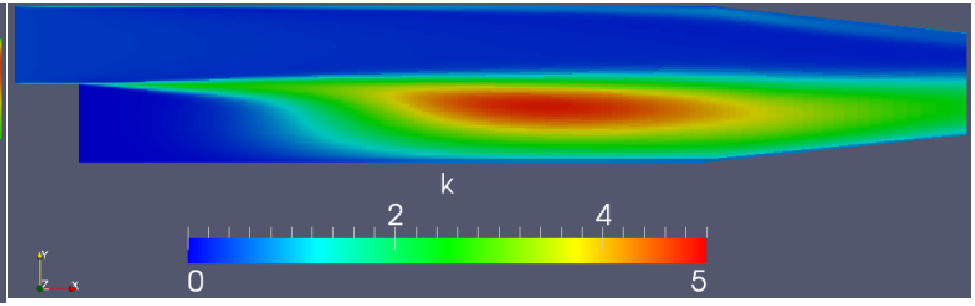


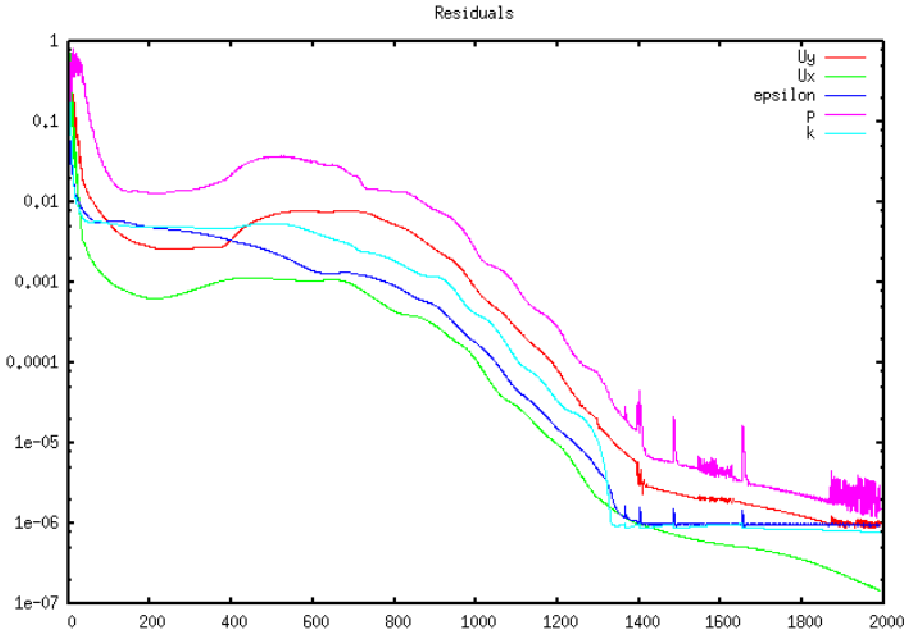


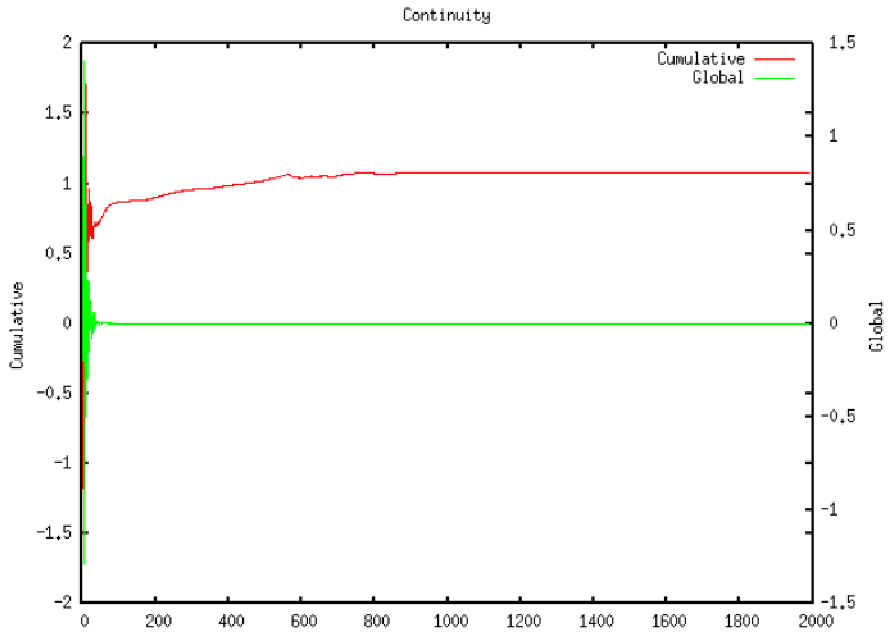
８. 計算結果の比較

No34　realizableKE,upwind,GAMG smoothSolver



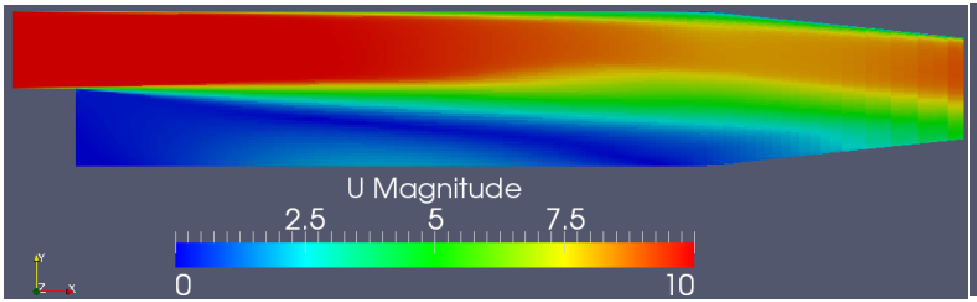


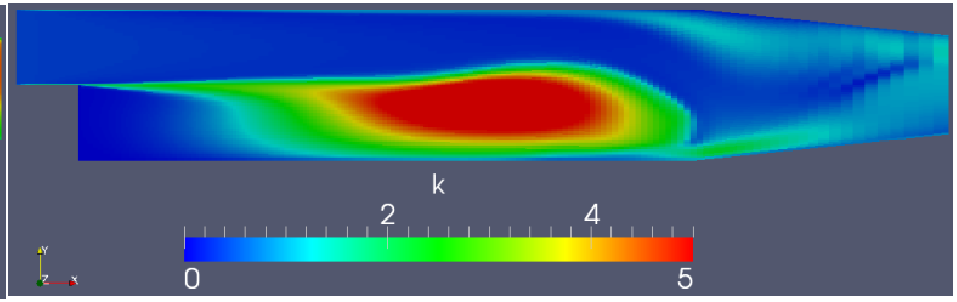


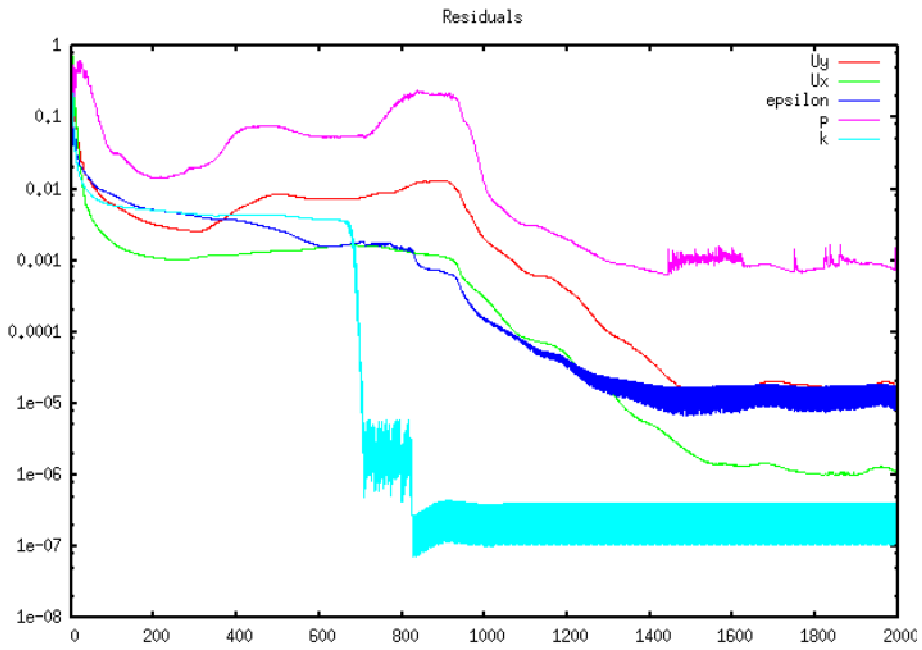


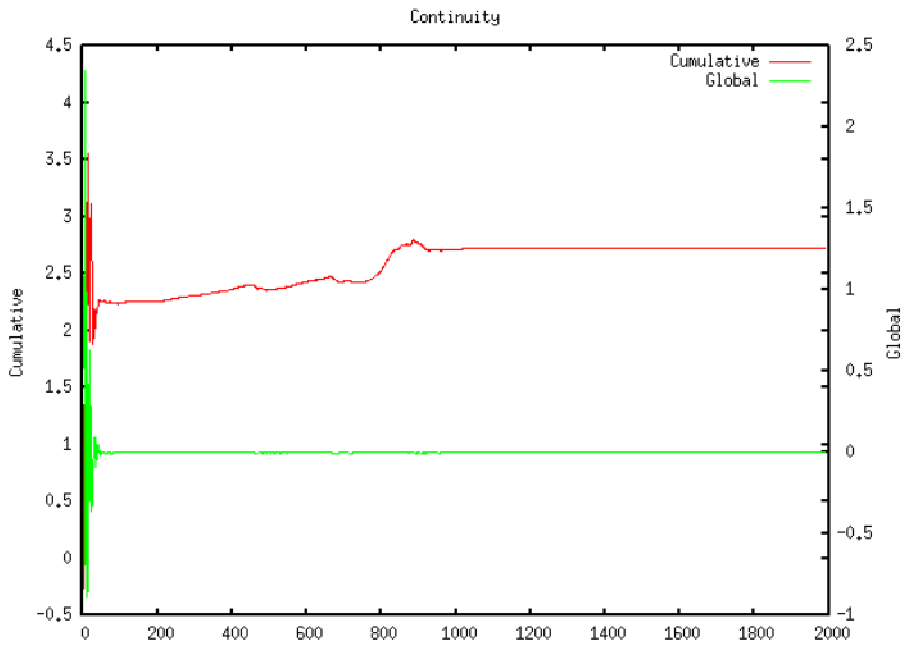
８. 計算結果の比較

No35　realizableKE,linear,GAMG smoothSolver



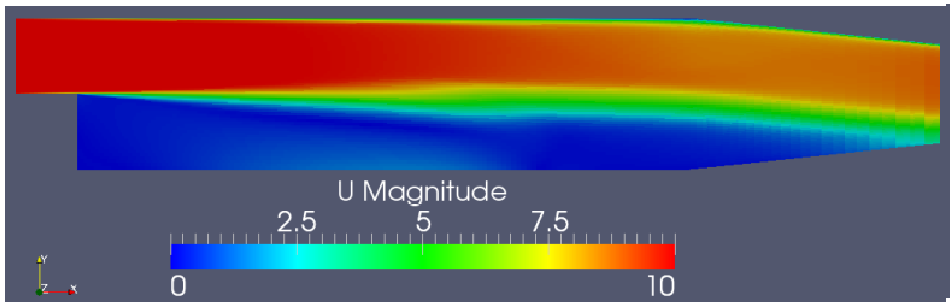


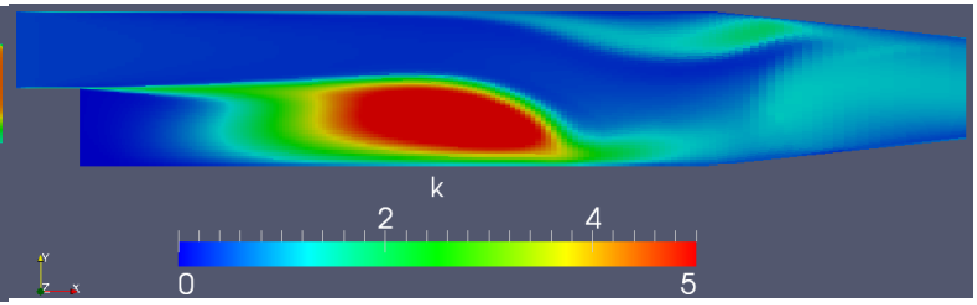


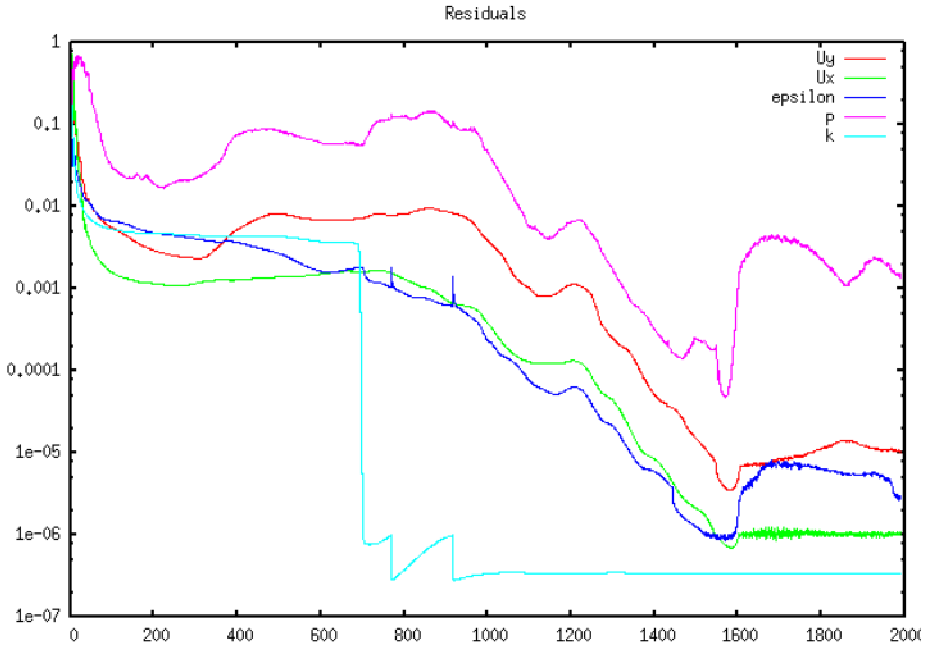


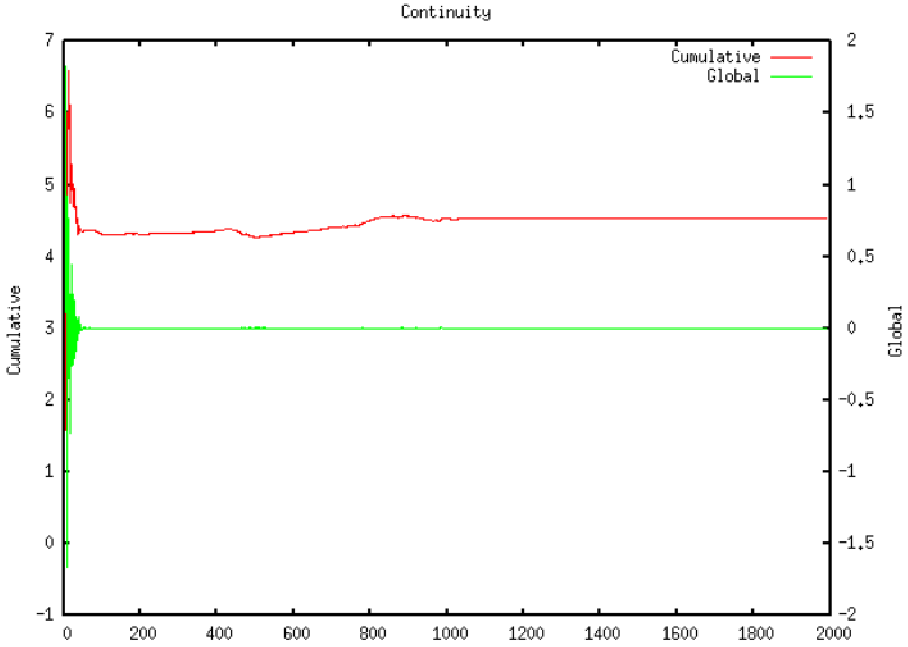
８. 計算結果の比較

No36　realizableKE,TVD,GAMG smoothSolver









８．計算結果の比較

今回の問題では次のような適用結果になった。

・fvSolutionでPCG,PBiCGを選んだ場合，fvSchemesでlinearはどの乱流モデルでも計算出来ない。数値粘性がないためと思われる。

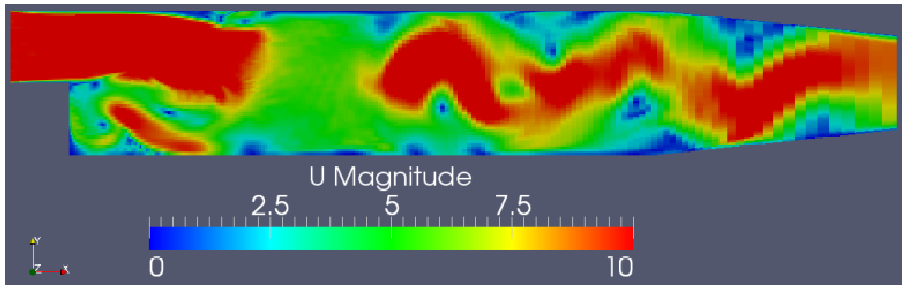
・fvSolutionでPCG,PBiCGを選んだ場合,kOmegaSSTでは計算出来ない。

・計算速度は，GAMG,smoothSolverがPCG,PBiCGより速い傾向にある。

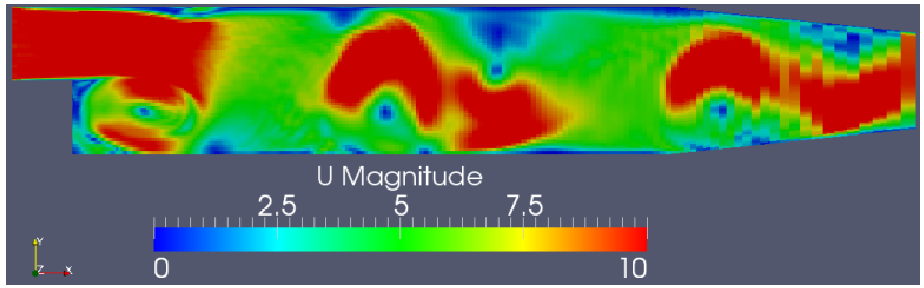
fvSolution	PCG,PBiCG			GAMG,smoothSolver		
fvSchemes	upwind	linear	TVD	upwind	linear	TVD
kEpsilon	○	×	○	○	×	○
kOmegaSST	×	×	×	○	○	○
realizableKE	○	×	○	○	○	○

９．LESでの計算　tutorials/incompressible/pisoFoam/les/pitzDaily

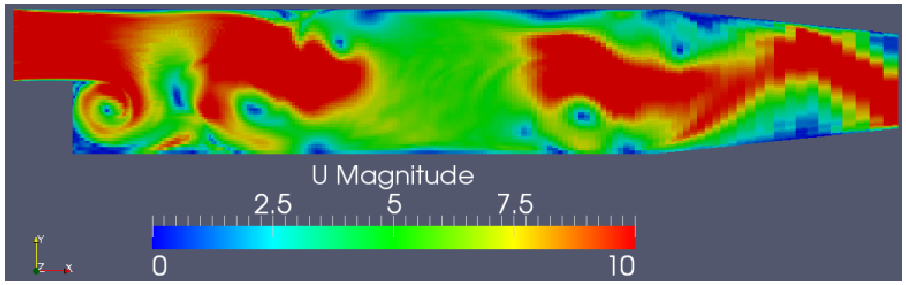
RANS系モデルとLESを比較してみた。えらく結果が異なる。
さあ，どう結果を理解すれば良いものやら？



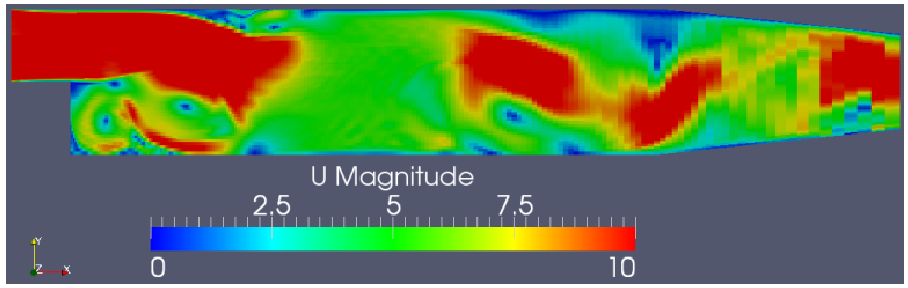
0.07s



0.09s



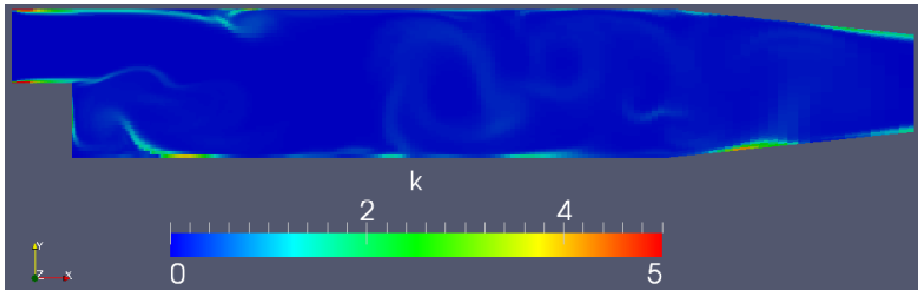
0.08s



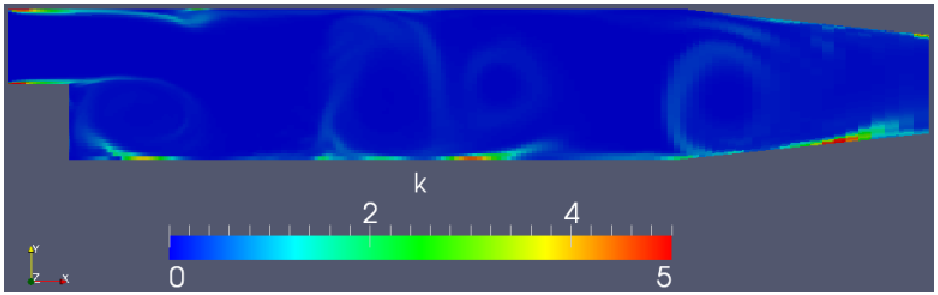
0.10s

９．LESでの計算　tutorials/incompressible/pisoFoam/les/pitzDaily

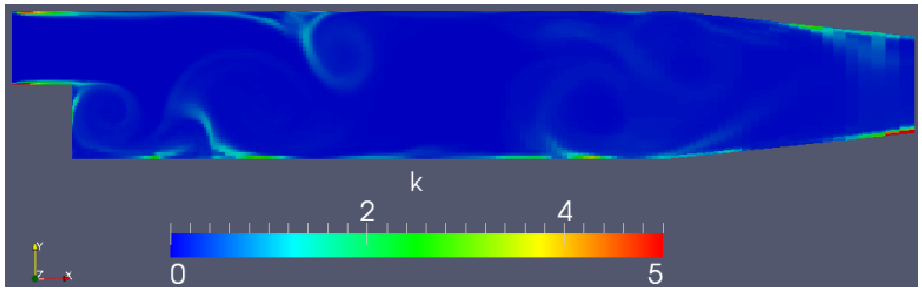
kの分布は速度場以上にRANSの解析結果と異なる。これもどうを理解すれば良いものやら？



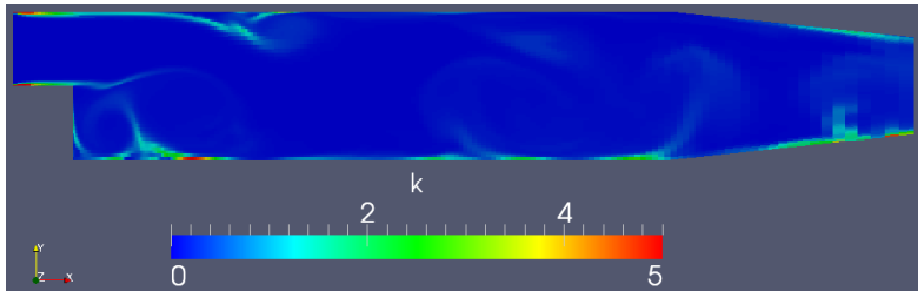
0.07s



0.09s



0.08s



0.10s

９．LESでの計算

この事例では，計算時間がRANS系に比べて10倍程度かかる。
1CUP=1717sec

2並列で計算してみると，1330sec（77%）

4並列で計算してみると，1562sec（91%）　逆に長くなった

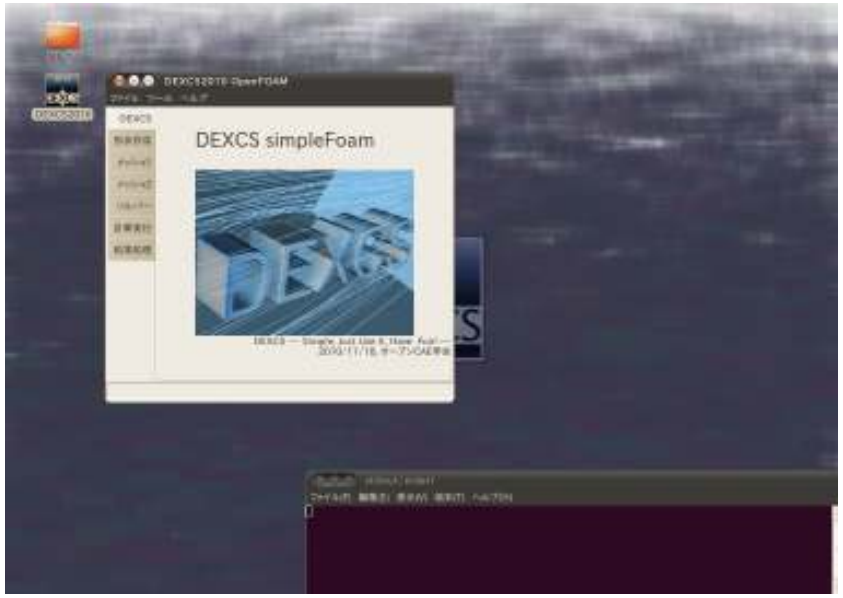
メッシュ数が少ないから並列化の効率は上がらなかった。

１０．DEXCS 2010を利用したバックステップ流れの操作

（１）DEXCS2010の立ち上げ
（２）新規解析ホルダの設定
（３）チュートリアルファイルの設定
（４）blockMeshの実施
（５）constantホルダファイルの編集
（６）Field変数の編集
（７）計算実行
（８）結果処理

（1）DEXCS2010の立ち上げ

ダブルクリックする



ランチャーが立ち上がってくる

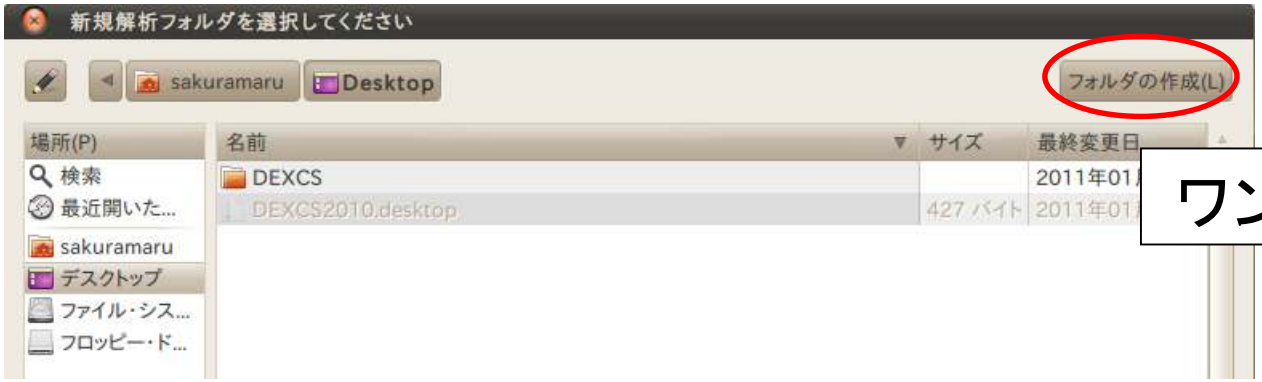
（2）新規解析ホルダの設定

（2）新規解析ホルダの設定

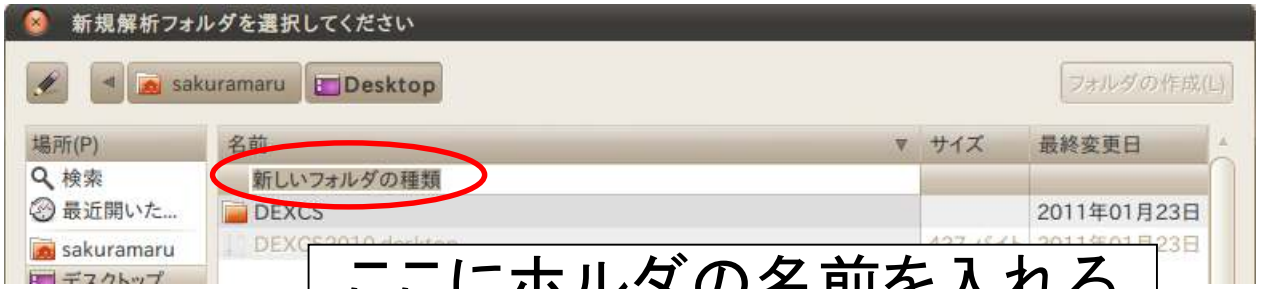
Desktop上にホルダを作成したいため，Desktopをダブルクリックする

Desktop上に移動した

（2）新規解析ホルダの設定

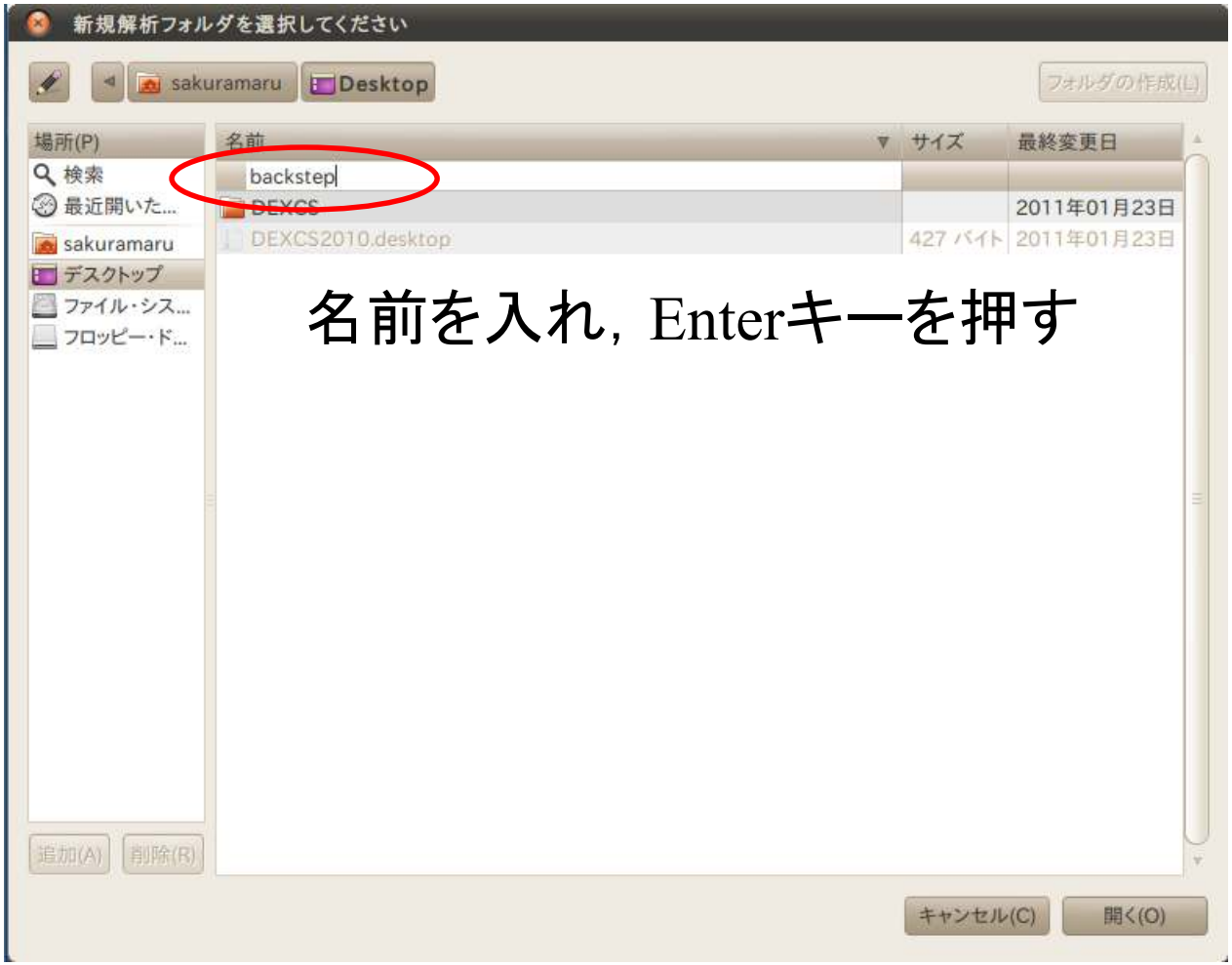


ワンクリックする



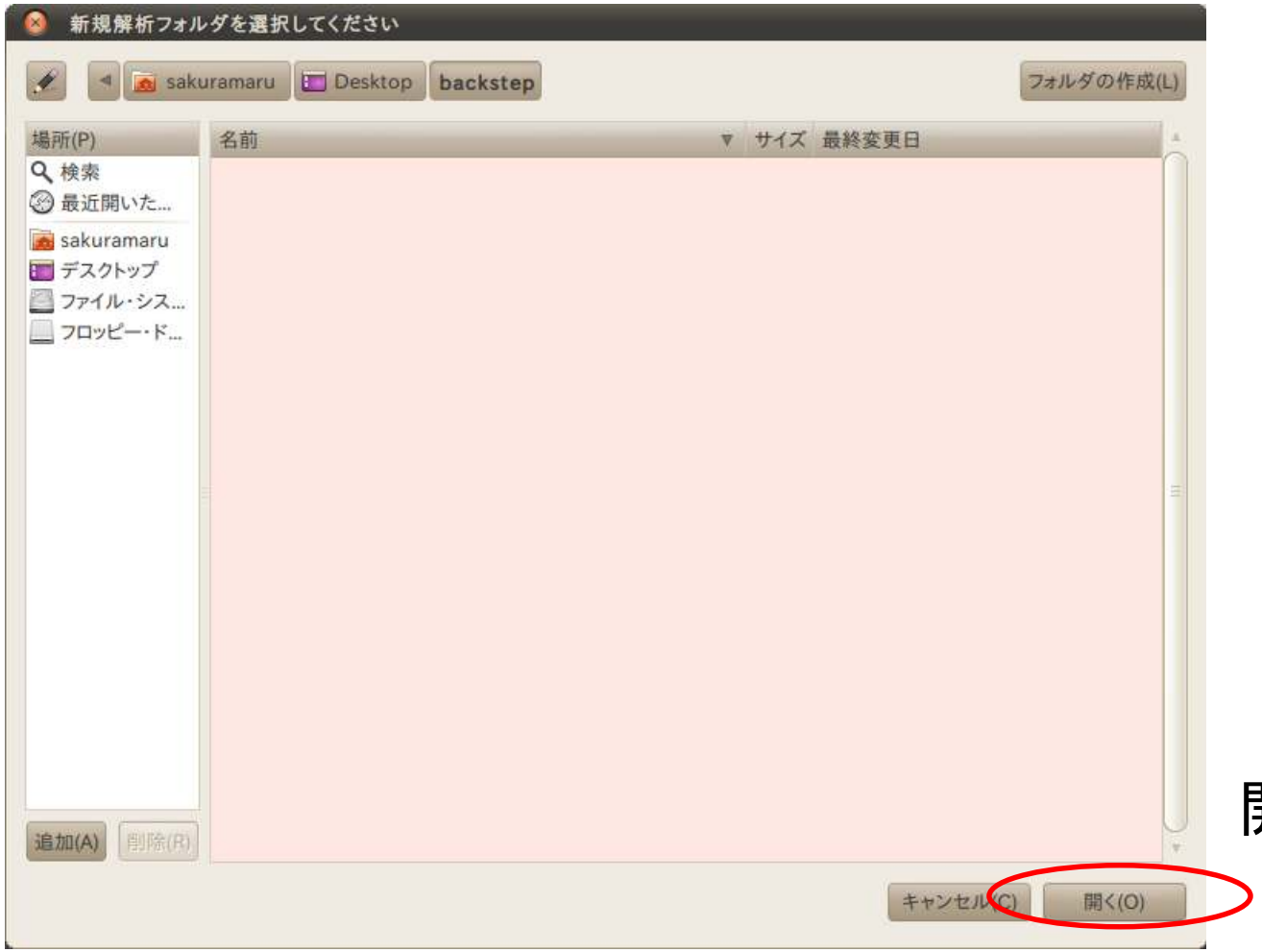
ここにホルダの名前を入れる
今回はbackstepとする

（2）新規解析ホルダの設定



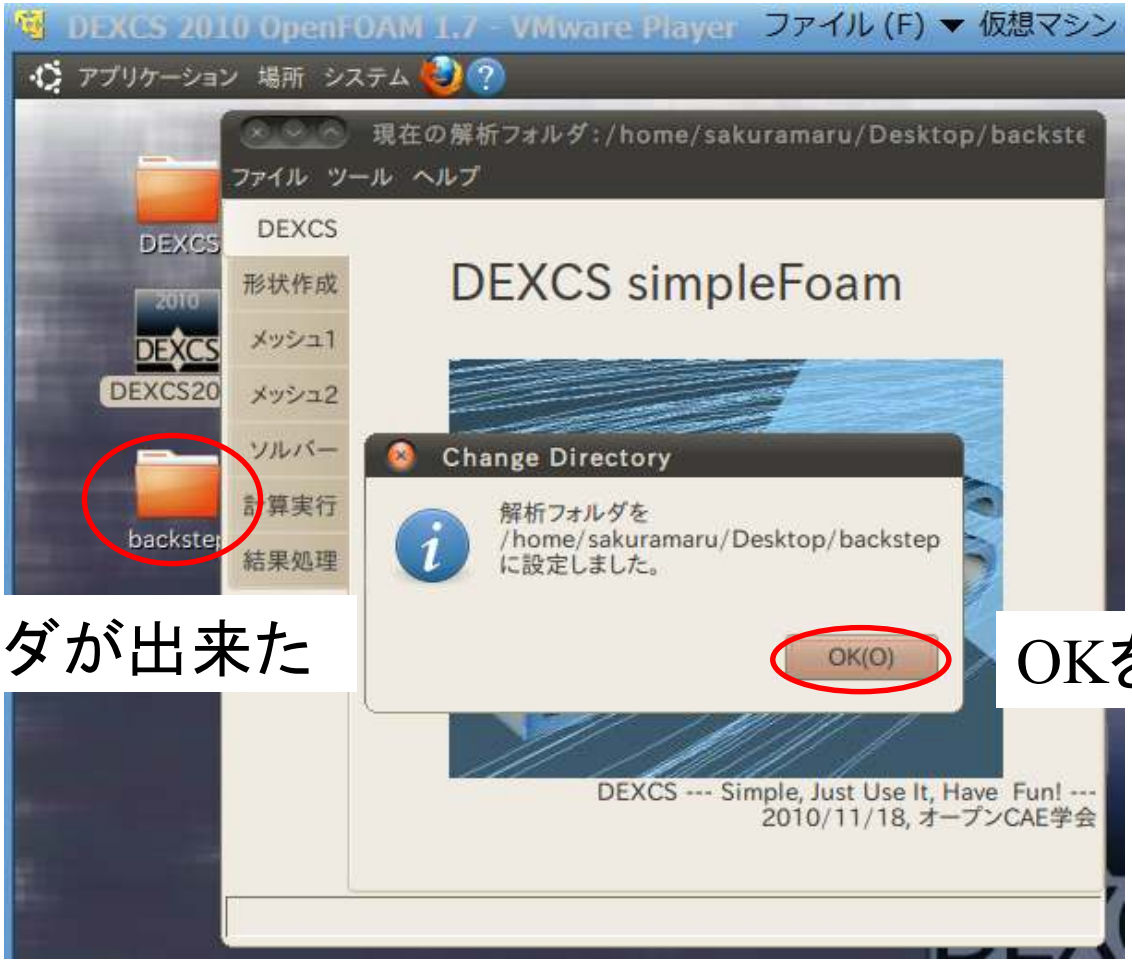
名前を入れ，Enterキーを押す

（2）新規解析ホルダの設定



開くを押す

（2）新規解析ホルダの設定

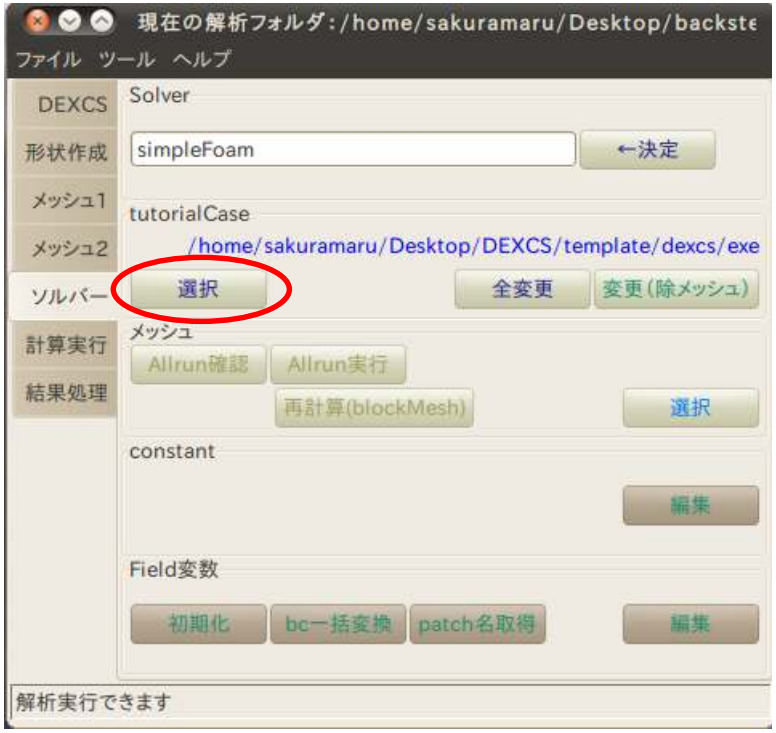


ホルダが出来た

OKを押す

（3）チュートリアルファイルの設定

選択を押す



（3）チュートリアルファイルの設定

（3）チュートリアルファイルの設定

（3）チュートリアルファイルの設定

pitzDailyをワンクリックする

pitzDailyが選択されると色が着く

開くを押す

（3）チュートリアルファイルの設定

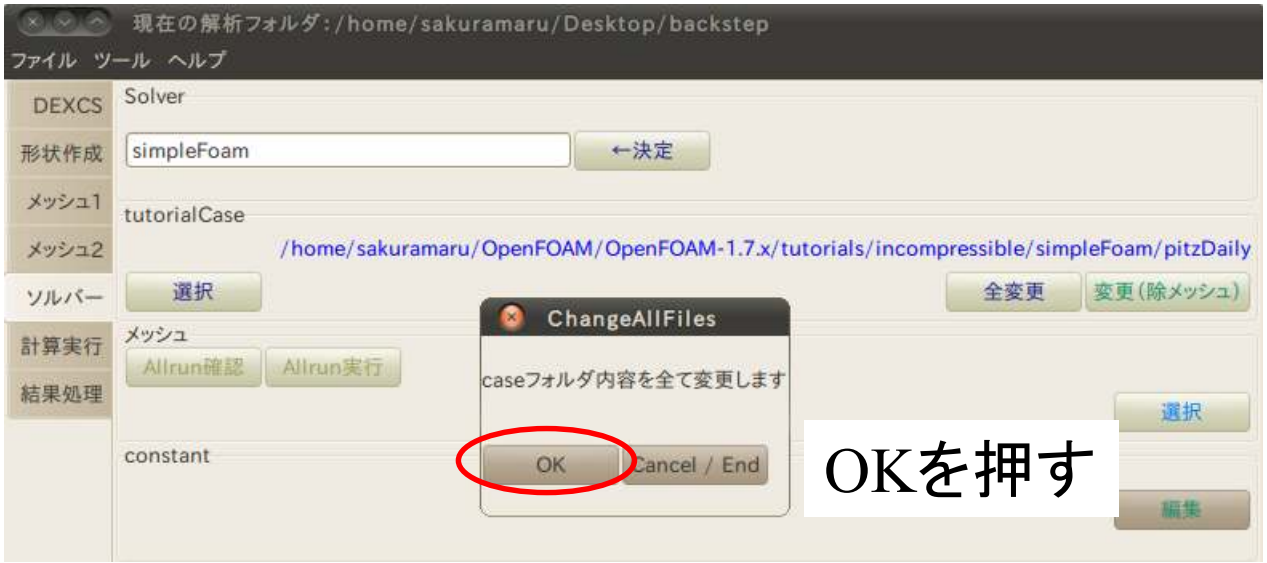
OKを押す

フォームの幅をマウスで広げるとpitzDailyが設定された事が確認できる

（3）チュートリアルファイルの設定

全変更を押す

（3）チュートリアルファイルの設定



OKを押す

処理状況が示される

（4）blockMeshの実施

再計算を押す

処理状況が示される

（5）constantホルダファイルの編集

編集を押す

OKを押す

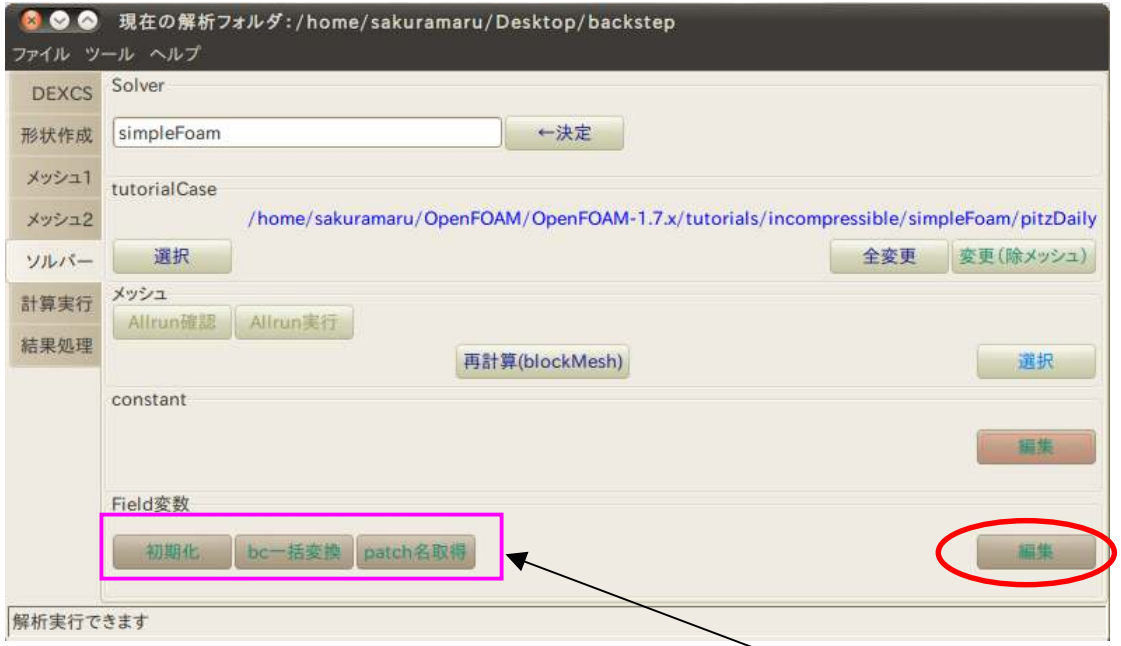
（5）constantホルダファイルの編集

geditが立ち上がり，ファイルがオープンされる

今回は何も変更しない

左上の×を押してgeditを終了する

（6）Field変数の編集



編集を押す

今回は使わない

OKを押す

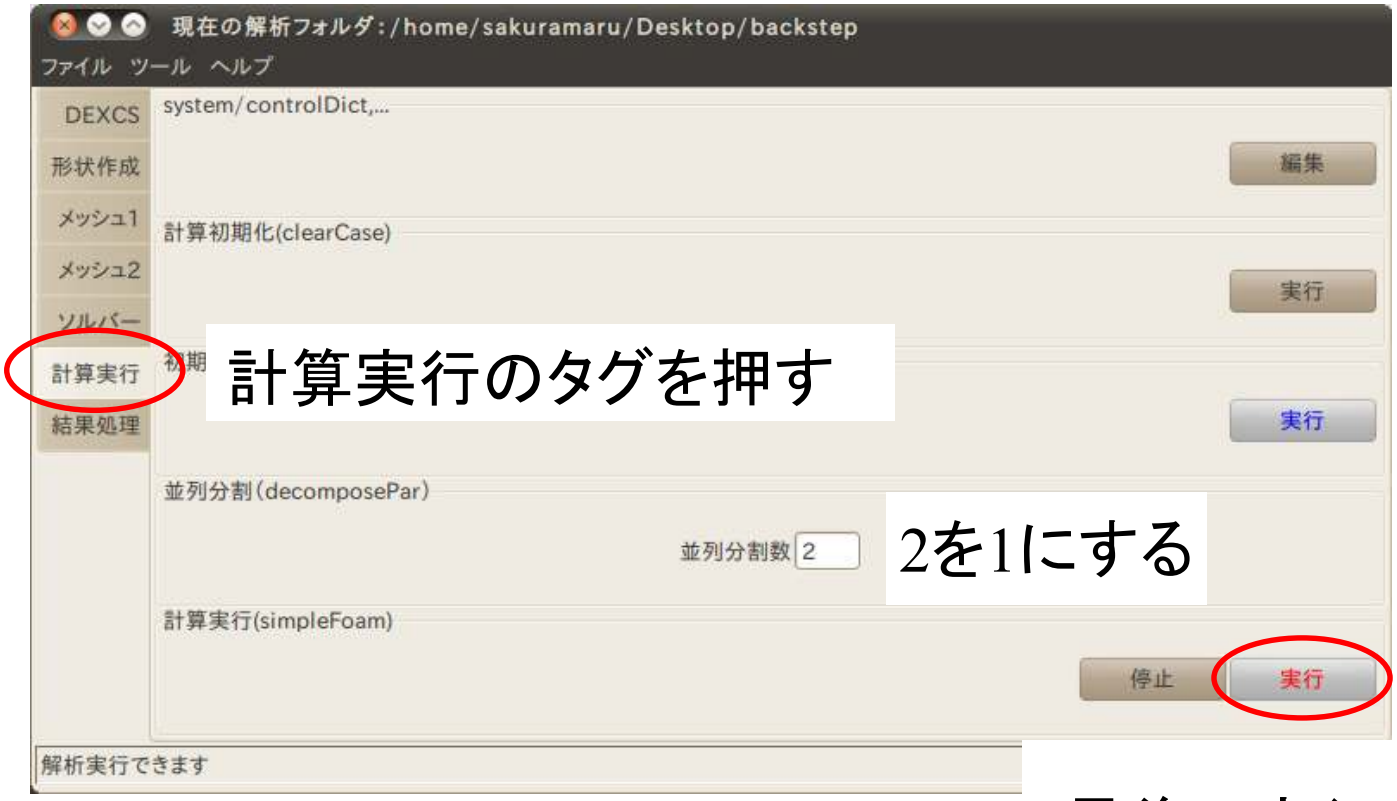
（6）Field変数の編集

geditが立ち上がり，ファイルがオープンされる

今回は何も変更しない

左上の×を押してgeditを終了する

(7)計算実行



計算実行のタグを押す

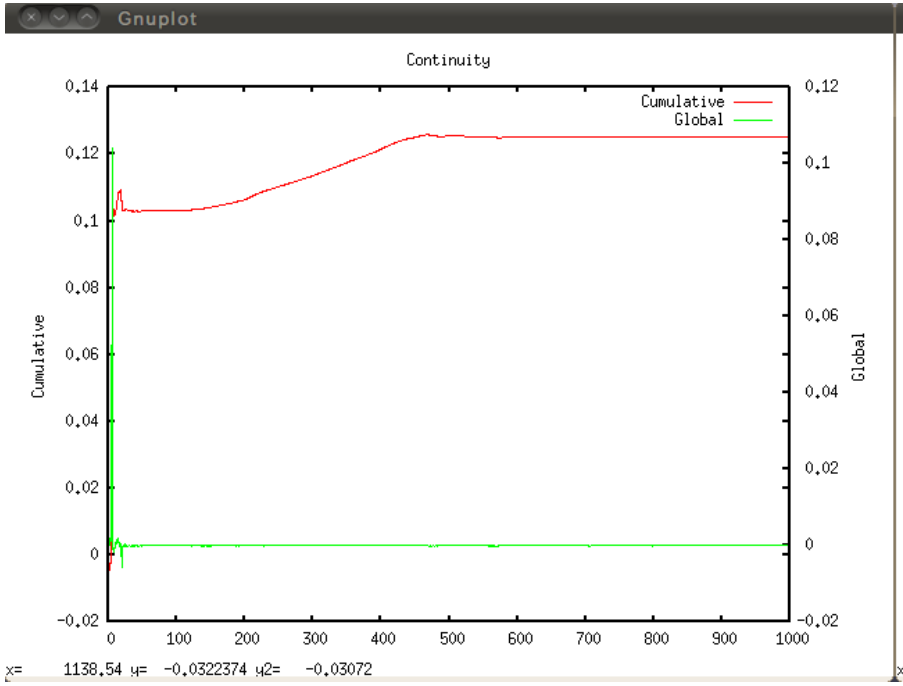
2を1にする

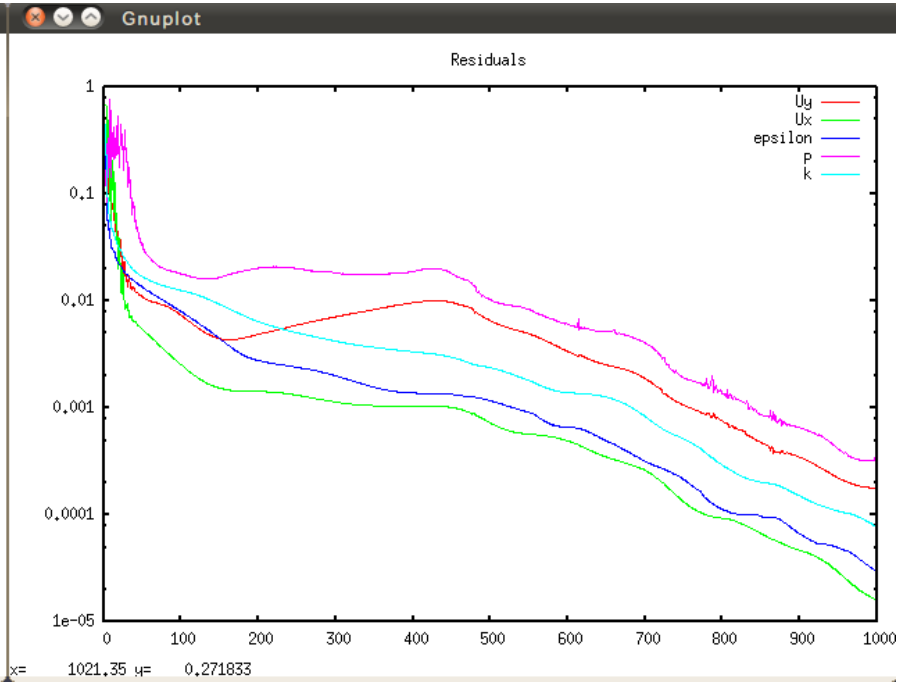
最後に実行を押す

(7)計算実行

OKを押すと計算が始まる

収束グラフが表示される





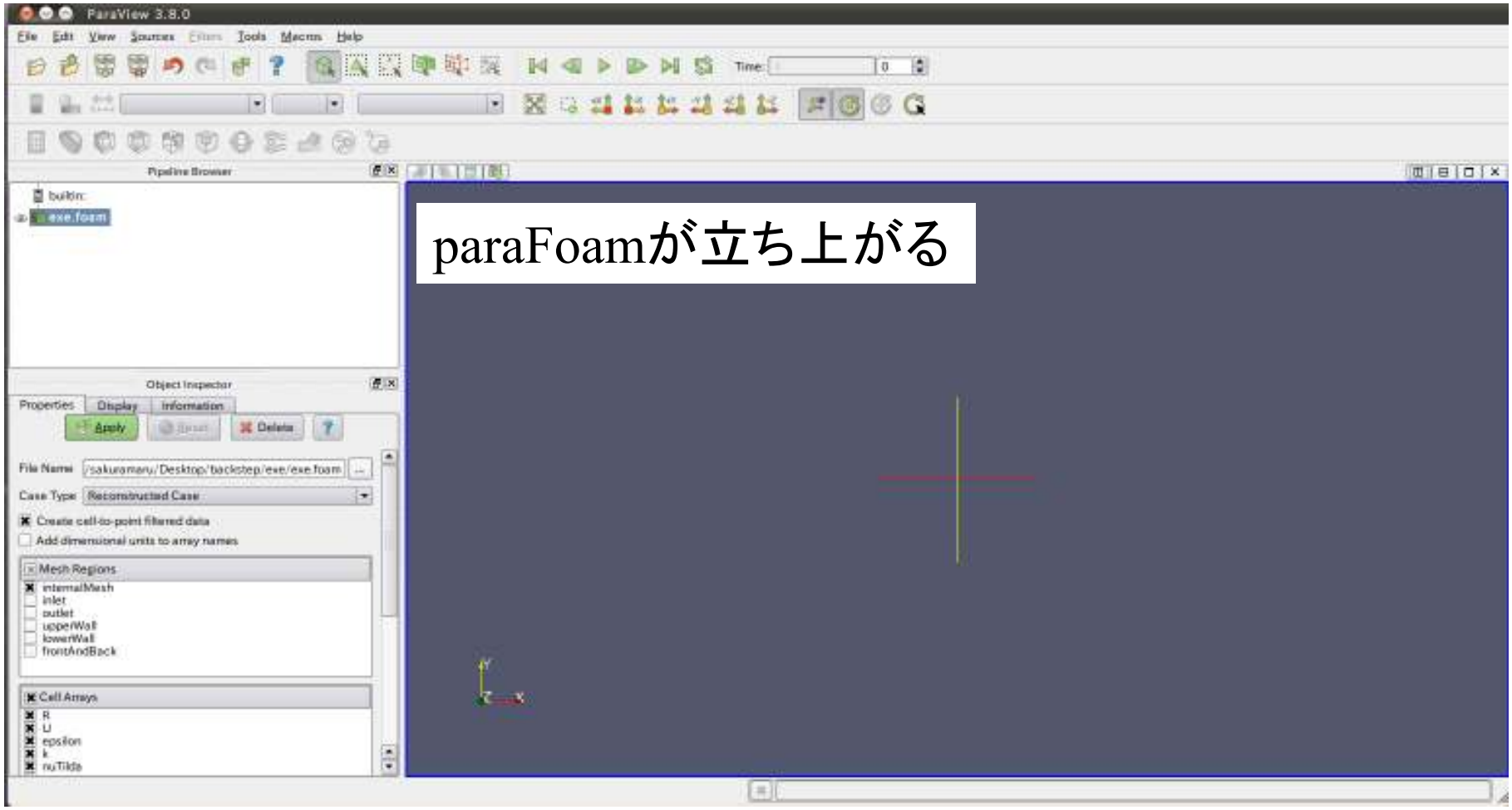
(8)結果処理


現在の解析フォルダ:/home/sakuramaru/Desktop/backstep
ファイル　ツール　ヘルプ
DEXCS
形状作成
メッシュ1
メッシュ2
ソルバー
計算実行
結果処理
paraFoam
paraFoamのタグを押す
実行
reconstruct
-latestTime
実行
foamLog
PyFoamRunner.simpleFoam.logfile
実行
結果処理のタグを押す
merge_probes.py 編集
実行
sample
sampleDict編集
実行
gnuplot
*.gp編集
実行
解析実行できます

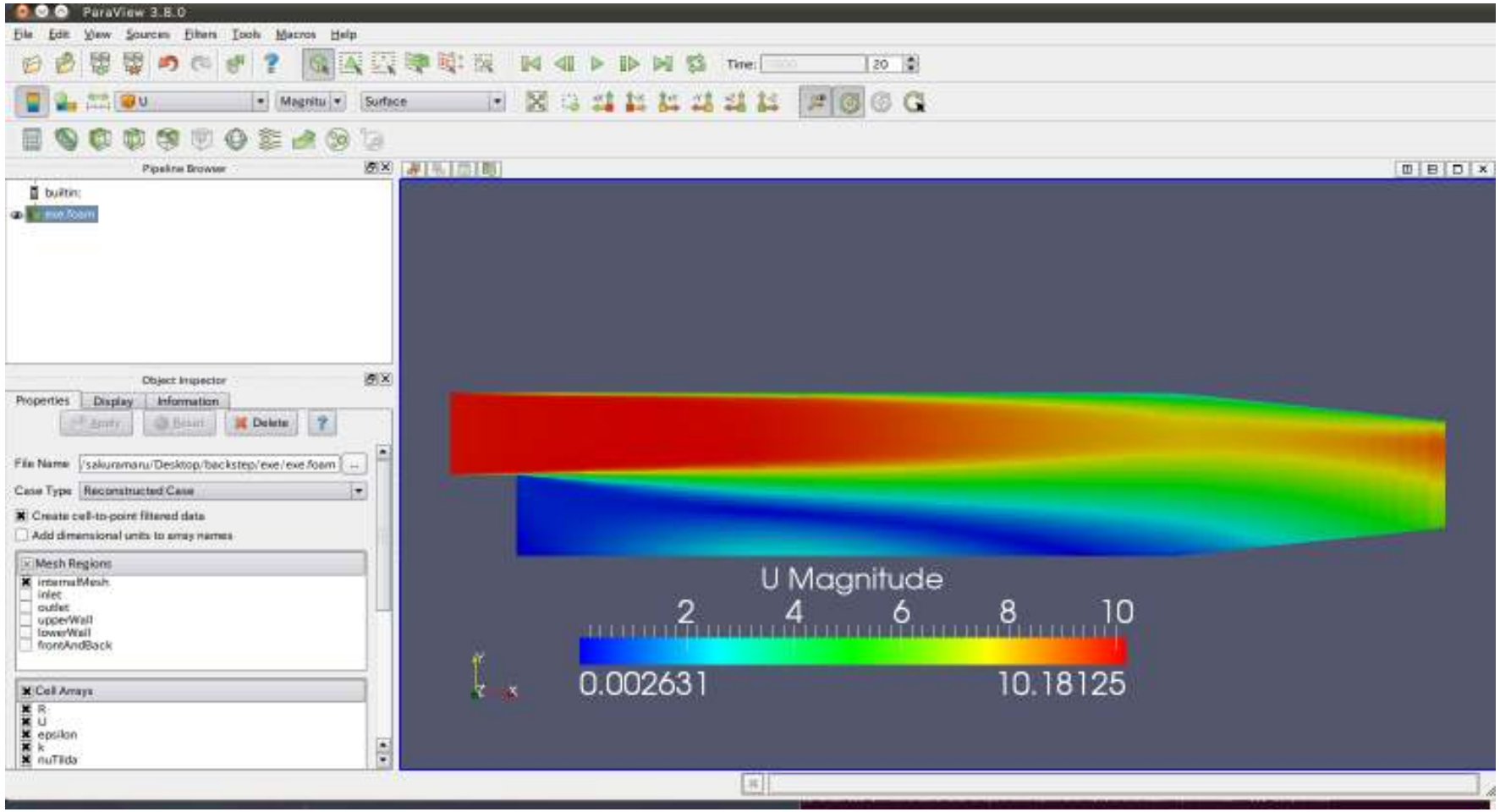
(8)結果処理


現在の解析フォルダ:/home/sakuramaru/Desktop/backstep
ファイル　ツール　ヘルプ
DEXCS
形状作成
メッシュ1
メッシュ2
ソルバー
計算実行
結果処理
paraFoam
実行
reconstruct
-latestTime
実行
foamLog
PyFoamRunner.simpleFoam.logfile
実行
merge_probes
merge_probes.py 編集
実行
sample
sampleDict編集
実行
gnuplot
*.gp編集
実行
RunParaFoamExe
paraFoam を実行します
OK
Cancel / End
OKを押す
解析実行できます

（8）結果処理


ParaView 3.8.0
File Edit View Sources Filters Tools Macros Help
Time:
Pipeline Browser
builtin:
exe.foam
Object Inspector
Properties Display Information
Apply Reset Delete
File Name /sakuramaru/Desktop/backstep/exe/exe.foam
Case Type Reconstructed Case
Create cell-to-point filtered data
Add dimensional units to array names
Mesh Regions
internalMesh
inlet
outlet
upperWall
lowerWall
frontAndBack
Cell Arrays
R
U
epsilon
k
nuTilda
paraFoamが立ち上がる

（8）結果処理


ParaView 3.8.0
File Edit View Sources Filters Tools Macros Help
Pipeline Browser
builtin:
exe.foam
Object Inspector
Properties Display Information
Apply Reset Delete
File Name /sakuramaru/Desktop/backstep/exe/exe.foam
Case Type Reconstructed Case
Create cell-to-point filtered data
Add dimensional units to array names
Mesh Regions
internalMesh
inlet
outlet
upperWall
lowerWall
frontAndBack
Cell Arrays
R
U
epsilon
k
nuTilda
U Magnitude
2 4 6 8 10
0.002631
10.18125

１１．参考資料

①www.cfd-online.com
②OpenFOAM ユーザガイド和訳Version 1.7.1
③OpenFOAM Programmer’s Guide Version 1.7.1
④平成22年度OpenFOAM非圧縮性流体解析演習シリーズ　第1回-第4回
⑤CFD of Air Flow in Hydro Power Generators（Pirooz Moradnia）
⑥CDAJ数値解析アカデミー　理論講座　乱流基礎コース
⑦CDAJ数値解析アカデミー　理論講座　CFD基礎コース

質疑・応答

ご清聴ありがとうございます。

アドバイスをよろしくお願いします。

（内容で間違っている部分があるかもしれません。初心者のためご容赦ください）