OpenFOAM 솔버 수렴특성

(Convergence characteristics of OpenFOAM solvers)

김 종태 (ex-kjt@kaeri.re.kr), 김 형태
한국원자력연구원 중대사고중수로안전연구부

2016 5th OpenFOAM Korea Users' Community Conference
2016. 9.29
한국해양대학교

Contents

- **Introduction**
  - Research background
  - OpenFOAM solver algorithm:
  - Solver modification

- **Convergence characteristics**
  - Numerical test of the solvers

- **summary**
  - Let's discuss what we do

Moderator circulation flow

- Buoyant and momentum forces acting on the fluid flow
  - Moderator flow around horizontal bundles inside an calandria tank of CANDU-6 reactors
  - Flow injection through inlet nozzles and volumetric heat source

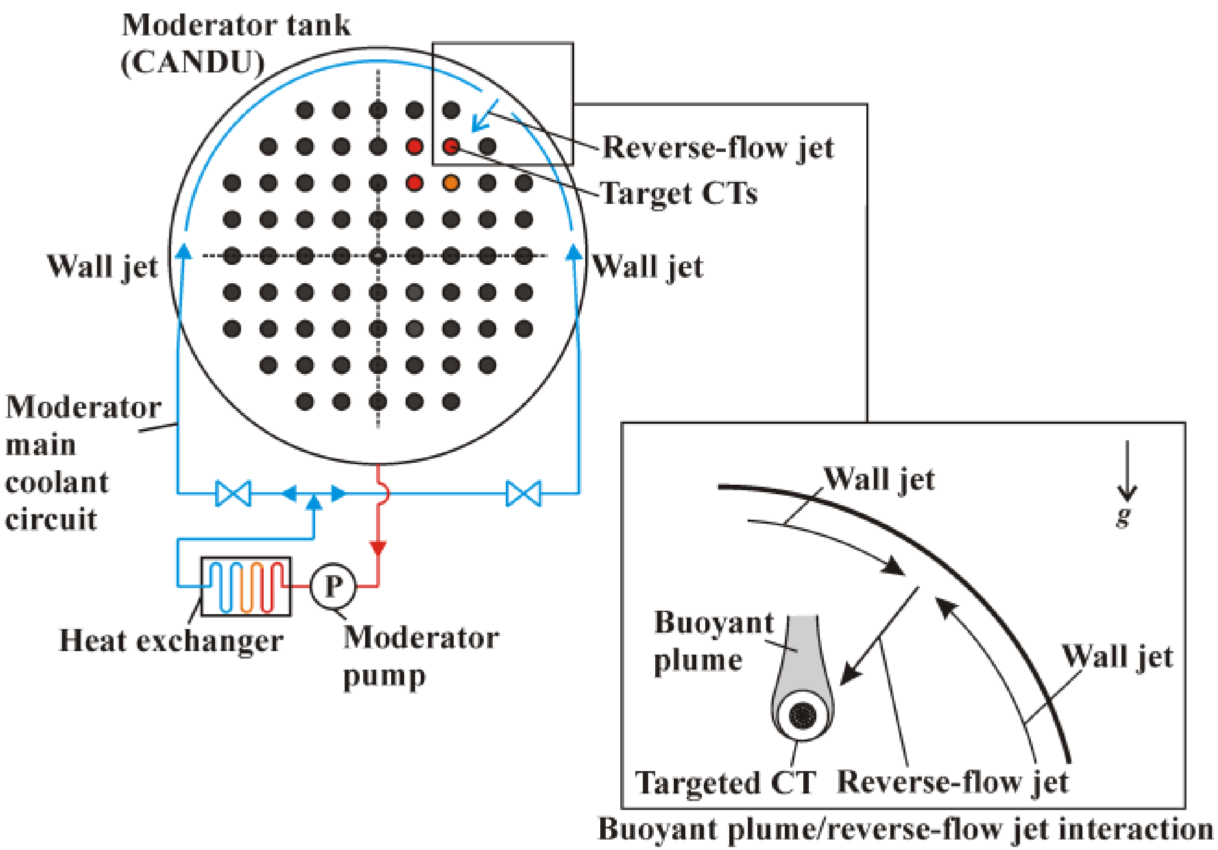
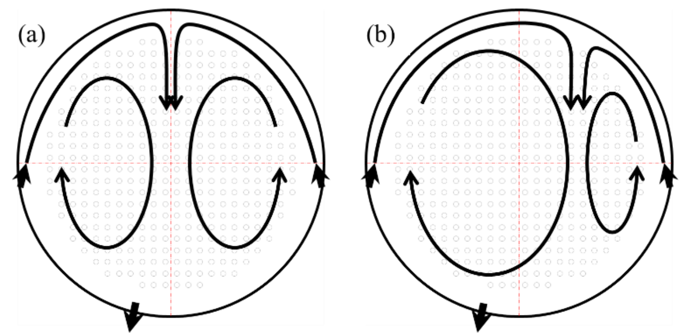


Fig. 1 Moderator circulation flow of CANDU-6

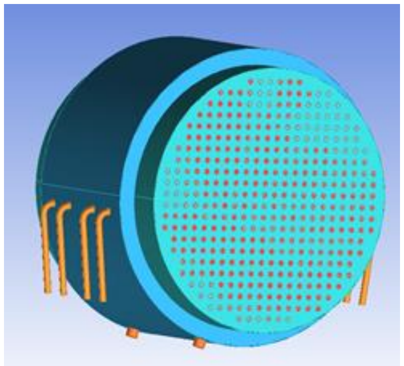


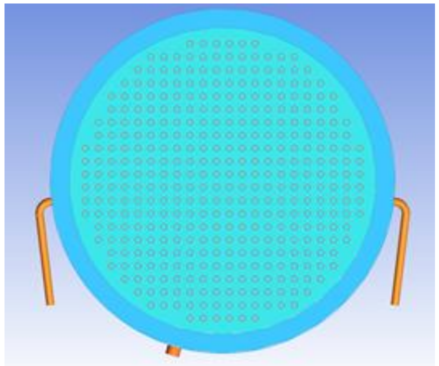
(a) Momentum-dominated flow (b) mixed flow

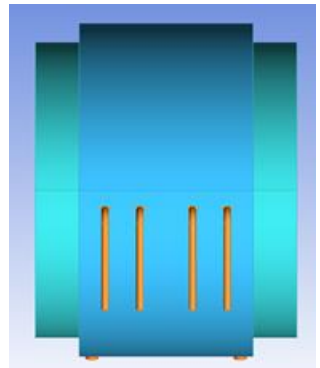
Fig. 2 Flow patterns of the moderator circulation

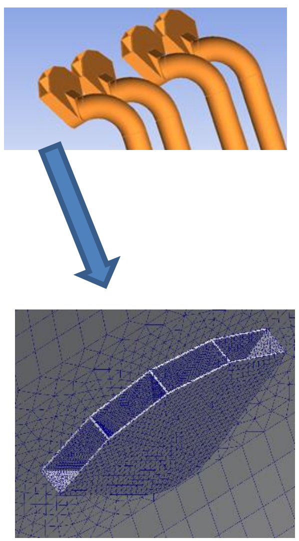
Moderator circulation flow

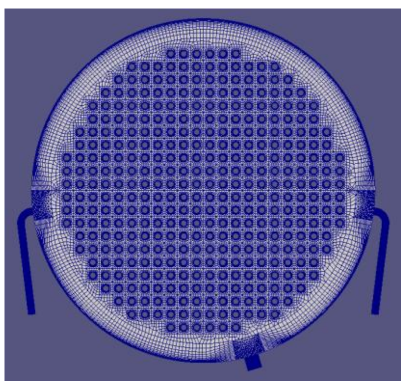
- Grid generation from geometry











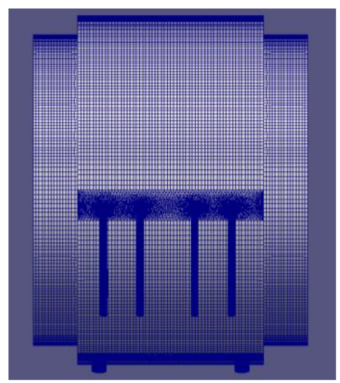


Table 1. Water properties

Parameter	Value
T (Inlet temperature)	47.3 °C
ρ	1085 kg/m^3
β	5×10^{-4} K^{-1}
μ	5.5×10^{-4} kg/m-s
c_p	4,207 J/kg-K
k	0.659 W/m-K

Total # of elements: 6,740,446
- Hexahedron: 5,112,270
- Pyramid: 13,112
- Triangular: 1,615,064

Moderator circulation flow

- Total Thermal Power = 100 MW
  - Radial and axial power distribution

$$Q_s(r,z) = Q_s\, f_r(r) f_z(z)$$

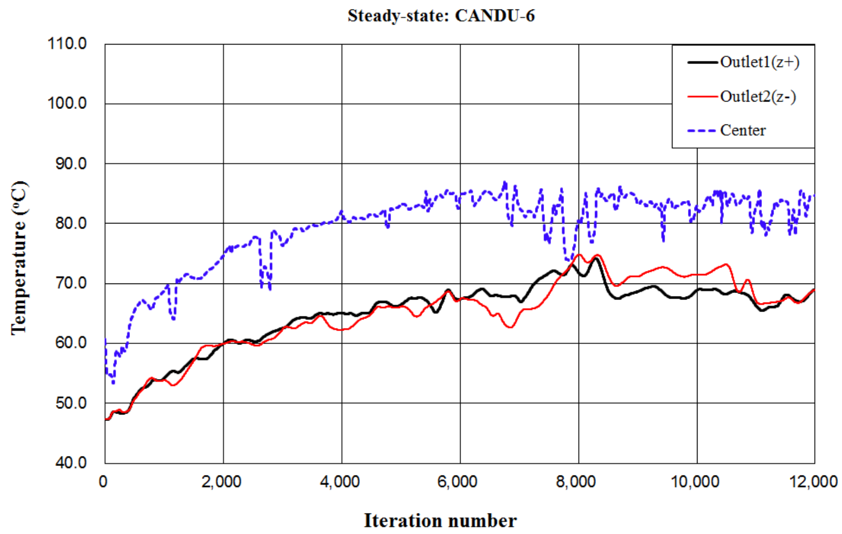
$$f_r(r) = 0.94588 - 0.01989\,r + 0.0995\,r^2 - 0.03888\,r^3 - 0.00256\,r^4 \; (0.0 \le r \le 3.38\,m)$$

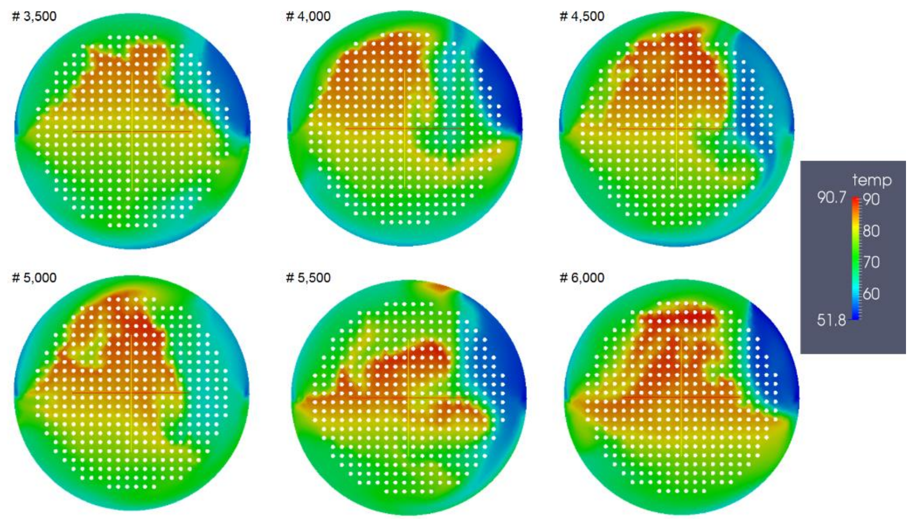
$$f_z(z) = 1.0 - 0.1111\,z^2 \; (-3.0 \le z \le 3.0\,m)$$

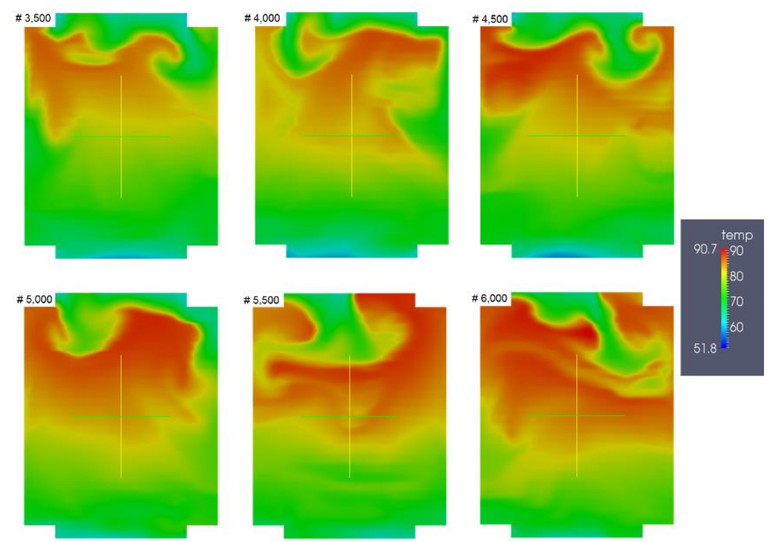
- Boundary/initial conditions
  - Mass flow rate at each inlet = 127.4 kg/s (Total Massflow = 1,019 kg/s)
  - No-slip at the wall
  - Outlet pressure: fixed, & zero-gradient otherwise.
  - Inlet temperature = 47.3 degC
  - Adiabatic at the wall
  - Initially stationary & isothermal.

Moderator circulation flow

- Reference calculation with default options







- **Basics solution algorithm of the OpenFOAM solvers**
  - ✓ Incompressible momentum equation

$$\frac{\partial}{\partial t}(\mathbf{U})+\nabla\cdot(\mathbf{U}\vec{U})+\nabla\cdot\mathbf{T}=-\nabla p+\mathbf{S}$$

  - ✓ Discretized momentum equation

$$A_U\mathbf{U}=\mathbf{H}(\mathbf{U})-\nabla p \quad \Leftarrow \quad \mathbf{H}(\mathbf{U})=-\sum_n A_n\mathbf{U}_n+\mathbf{S}_u$$

  - ✓ predictor step

$$\mathbf{U}^*=\frac{\mathbf{H}(\mathbf{U}^*)}{A_U}-\frac{\nabla p}{A_U} \qquad \phi^*=\mathbf{U}_f^*\cdot\vec{S}_f=\left(\frac{\mathbf{H}^*}{A_U}\right)_f\cdot\vec{S}_f-\left(\frac{1}{A_U}\nabla p\right)_f\cdot\vec{S}_f$$

  - ✓ Continuity equation

$$\nabla\cdot(\mathbf{U})=0$$

OpenFOAM solver algorithm (2)

- **Basics solution algorithm of the OpenFOAM solvers**
  - ✓ Corrector step

$$\phi = \mathbf{U}_f \cdot \vec{S}_f = \left(\frac{\mathbf{H}^*}{A_U} \right)_f \cdot \vec{S}_f - \left(\frac{1}{A_U} \nabla p \right)_f \cdot \vec{S}_f$$

$$\nabla \cdot (\mathbf{U}) = \sum_f \phi = 0 \quad \Rightarrow \quad \sum_f \left(\frac{\mathbf{H}(\mathbf{U}^*)}{A_U} \right)_f \cdot \vec{S}_f = \sum_f \left(\frac{\nabla p}{A_U} \right)_f \cdot \vec{S}_f$$

$$\mathbf{U} = \frac{\mathbf{H}(\mathbf{U}^*)}{A_U} - \frac{\nabla p}{A_U}$$

buoyantBoussinesqSimplefoam

- **Body force**
  - ✓ Momentum equation with body force
    - Boussinesq assumption for incompressible flow

momentum eq.

$$\frac{\partial}{\partial t}(\rho \mathbf{U}) + \nabla \cdot (\rho \mathbf{U}\vec{U}) + \nabla \cdot \rho \mathbf{T} = -\nabla p + \rho \mathbf{g} + \rho \mathbf{S}$$

density

$$\rho = \rho_0 - \rho_0 \beta (T - T_0) \Rightarrow \frac{\rho}{\rho_0} = 1 - \beta (T - T_0) = 1 + \rho' = \rho_k$$

apply boussinesq assumption

$$\frac{\partial}{\partial t}(\rho_0 \mathbf{U}) + \nabla \cdot (\rho_0 \mathbf{U}\vec{U}) + \nabla \cdot \rho_0 \mathbf{T} = -\nabla p + \rho \mathbf{g} + \rho_0 \mathbf{S}$$

divide by ρ_0

$$\frac{\partial}{\partial t}(\mathbf{U}) + \nabla \cdot (\mathbf{U}\vec{U}) + \nabla \cdot \mathbf{T} = -\nabla p' + \rho_k \mathbf{g} + \mathbf{S}$$

Body force issue

- **Body force**
  - ✓ Momentum equation with body force
    - Boussinesq assumption for incompressible flow

momentum eq. $$\frac{\partial}{\partial t}(\mathbf{U}) + \nabla \cdot (\mathbf{U}\vec{U}) + \nabla \cdot \mathbf{T} = -\nabla p + \rho_k \mathbf{g} + \mathbf{S}$$ ⟸ $p' \Rightarrow p$

pressure and body force $$-\nabla p + \rho_k \mathbf{g} = -\nabla(p - \rho_k g h) - g h \nabla \rho_k = -\nabla p_{rgh} - g h \nabla \rho_k$$

modified momentum eq. $$\frac{\partial}{\partial t}(\mathbf{U}) + \nabla \cdot (\mathbf{U}\vec{U}) + \nabla \cdot \mathbf{T} = -\nabla p_{rgh} - g h \nabla \rho_k + \mathbf{S}$$

discretised momentum eq. $$A_U \mathbf{U} = \mathbf{H}(\mathbf{U}) - \nabla p_{rgh} - g h \nabla \rho_k$$

Body force induced problem

- Body force is strongly coupled with pressure gradient
  - ✓ Especially in low-momentum flow such as a natural convection, pressure gradient from the body force is bigger
  - ✓ In some cases, flow near wall is contaminated because of its low momentum
    - solution problem
      - checker-boarding pressure field
      - decoupling of pressure and velocity field
    - remedy
      - body force added on cell faces for the non-staggered mesh method
      - body force weighted pressure interpolation
        - wall pressure extrapolation using body force

$$p_b = p_0 + \nabla p_0 \cdot d\vec{s}_0 \qquad \nabla p_0 = \rho \mathbf{g}$$

        - cell-face pressure interpolation *(Numerical Heat transfer B 68, 2015)*

$$p_f = \frac{1}{2}(p_l + p_r) \qquad p_l = p_i + \rho_i \mathbf{g} \cdot d\vec{r}_i \qquad p_r = p_j + \rho_j \mathbf{g} \cdot d\vec{r}_j$$

P_{rhg} in buoyantBoussinesqSimplefoam

- **Pressure**
  - ✓ Pressure changed by
    - hydrodynamic, hydrostatic, thermodynamic et al.
  - ✓ Pressure equation in buoyantBoussinesqSimpleFoam
    - hydrodynamic pressure

$$p_{rgh} = p - \rho_k g h$$

  - ✓ on wall, $\mathbf{U} \approx 0$, then momentum equation becomes
    - In general solver

$$-\nabla p + \rho_k \mathbf{g} = 0 \Rightarrow \nabla p \cdot \vec{S}_f = \frac{dp}{dn} S_f = \rho_k \mathbf{g} \cdot \vec{S}_f \Rightarrow \frac{dp}{dn} = \rho_k \mathbf{g} \cdot \hat{n}$$

    - In OpenFAOM

$$-\nabla p_{rgh} - gh \nabla \rho_k = 0 \Rightarrow \nabla p_{rgh} \cdot \vec{S}_f = -gh \nabla \rho_k \cdot \vec{S}_f \Rightarrow \frac{dp_{rgh}}{dn} = -gh \frac{d\rho_k}{dn}$$

      - implemented in the solver and fixedFluxPressureFvPatch

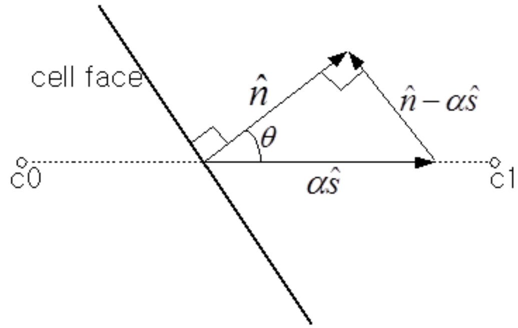
Convergence problem

- **Solution oscillating**
  - ✓ Physical oscillation ➔ must be captured
  - ✓ numerical oscillation ➔ must be removed
  - ✓ But sometimes it is difficult to distinguish the physical and numerical oscillation

- **Why not converged or oscillating?**
  - ✓ numerical schemes problem
  - ✓ mesh non-orthogonality problem
  - ✓ p-U/p-T-U coupling problem
  - ✓ initialization problem
  - ✓ physical model problem such as a turbulence model

Non-orthogonality correction in OpenFOAM

- **Need for non-orthogonality correction**
  - ✓ Cross diffusion of the pressure Laplacian term is increased



- **It is different from standard SIMPLE algorithm**
  - ✓ In standard SIMPLE algorithm pressure correction eq. is solved
  - ✓ Before solving p' eq. it is not known
- **OpenFOAM solves pressure equation**
  - ✓ Cross diffusion of the pressure Laplacian term can be calculated before solving the pressure equation

Modification of OpenFOAM solvers

- **Pressure relaxation**
  - ✓ Before or after momentum correction? (NextFoam 길재흥, 2015 OKUCC)
  - ✓ Many SIMPLE-based CFD codes do the p-relaxation after the velocity correction.

```
flux update
U update
pressure relax
```

  - ✓ OpenFOAM does the p-relaxation before the velocity update .

```
flux update ➔  phi = phiHbyA - p_rghEqn.flux();
p relax     ➔ p_rgh.relax();
U update    ➔ U = HbyA + rAU*fvc::reconstruct((phig - p_rghEqn.flux())/rAUf)
```

  - ✓ Modified solver

```
phi = phiHbyA - p_rghEqn.flux();
U = HbyA + rAU*fvc::reconstruct((phig - p_rghEqn.flux())/rAUf)
p_rgh.relax();
```

Improvement of pressure implicitness

- **Discretized momentum equation**

$$A_U \mathbf{U}^* = \mathbf{H}(\mathbf{U}^*) - \nabla p^m$$

✓ $\mathbf{U}^*$ is predicted value

✓ but, p^m is previous value. ⇨ semi-implicit

- **Additional pressure equation**

$$A_U \mathbf{U}^m = \mathbf{H}(\mathbf{U}^m) - \nabla p^*$$

✓ Before solving the momentum eq., construct pressure equation

$$\phi = \mathbf{U}_f^m \cdot \vec{S}_f = \left(\frac{\mathbf{H}(\mathbf{U}^m)}{A_U}\right)_f \cdot \vec{S}_f - \left(\frac{1}{A_U}\nabla p^*\right)_f \cdot \vec{S}_f \Rightarrow \sum_f \left(\frac{\mathbf{H}(\mathbf{U}^m)}{A_U}\right)_f \cdot \vec{S}_f = \sum_f \left(\frac{\nabla p^*}{A_U}\right)_f \cdot \vec{S}_f$$

✓ Now, we get a guessed pressure

$$A_U \mathbf{U}^* = \mathbf{H}(\mathbf{U}^*) - \nabla p^*$$

Numerical test

- **OpenFOAM version for numerical test**
  - ✓ OpenFOAM 2.4
  - ✓ OpenFOAM 3.0
  - ✓ OpenFOAM dev
- **OpenFOAM solvers for numerical test**
  - ✓ simpleFoam
  - ✓ buoyantBoussinesqSimpleFoam
  - ✓ buoyantBoussinesqPimpleFoam
  - ✓ buoyantSimpleFoam
  - ✓ buoyantPimpleFoam

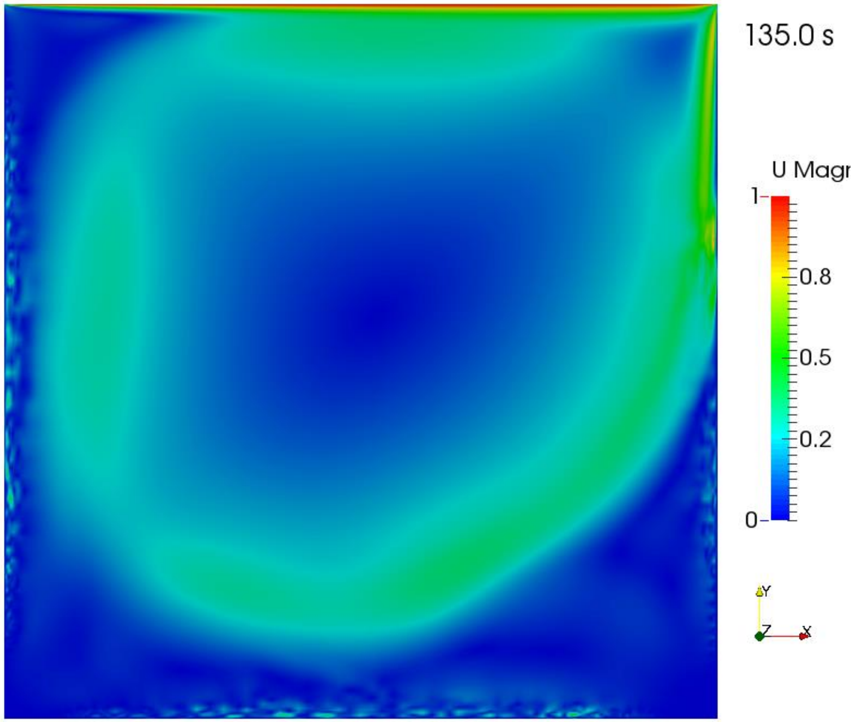
pimpleFoam

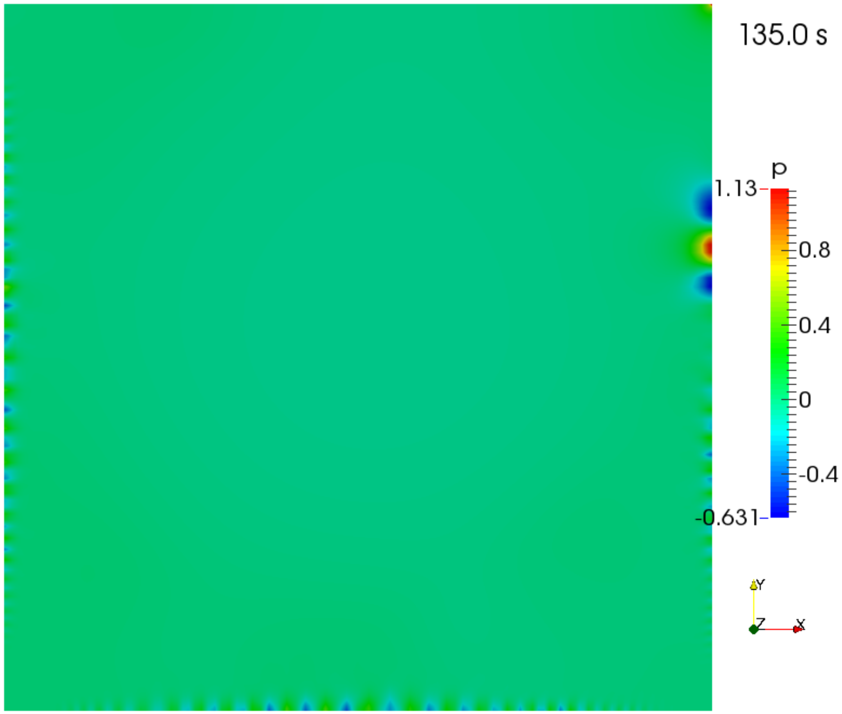
- **OpenFOAM version**
  - ✓ OpenFOAM dev
- **Solver modification**
  - ✓ additional pressure prediction before solving momentum eq.

- **Test case**
  - ✓ lid-driven cavity flow problem
    - Re = 10,000

Lid-driven cavity flow

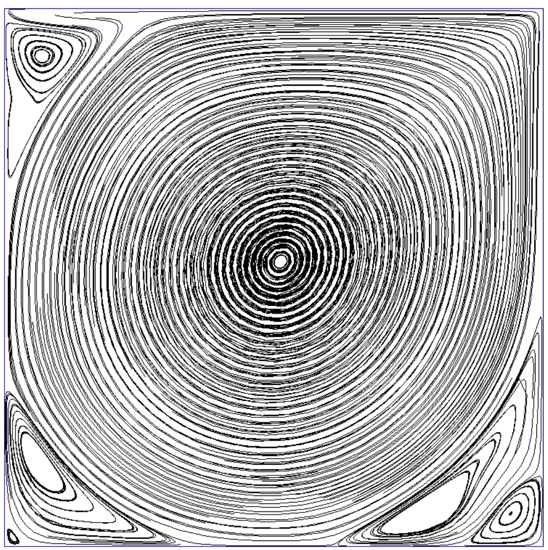
- **Solver: pimpleFoam**
- **max Co number 50**
  - ✓ run in PISO mode
- **checker-board in pressure and velocity fields**

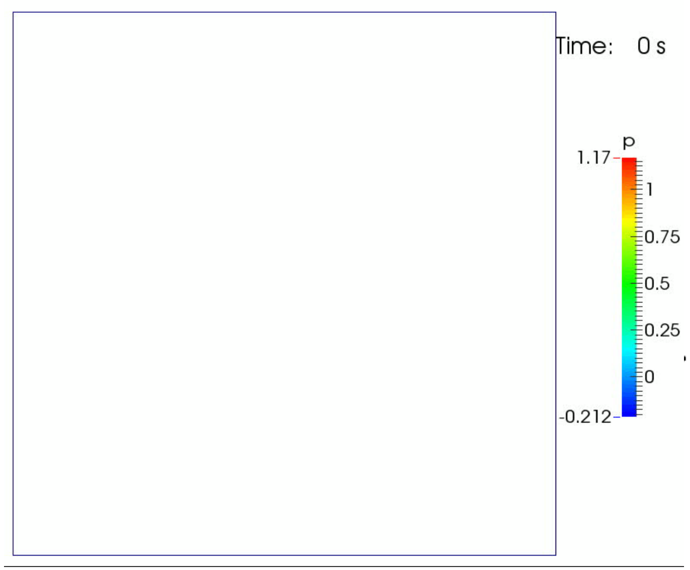


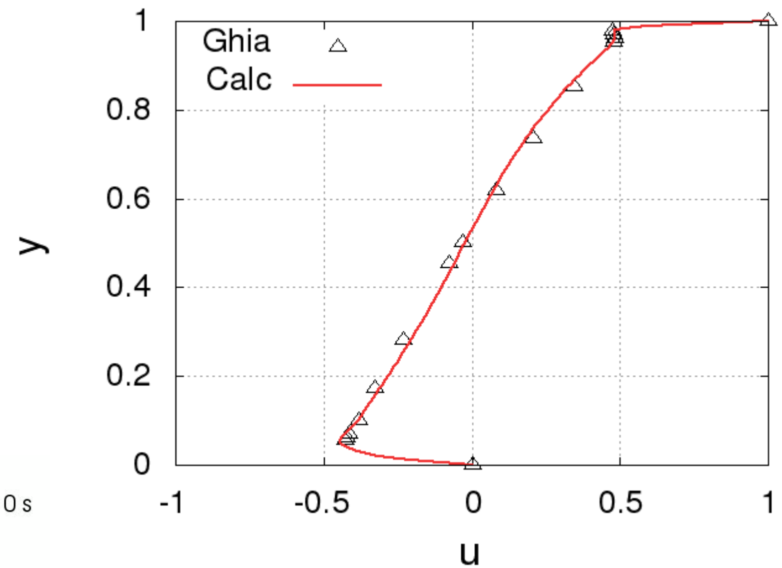


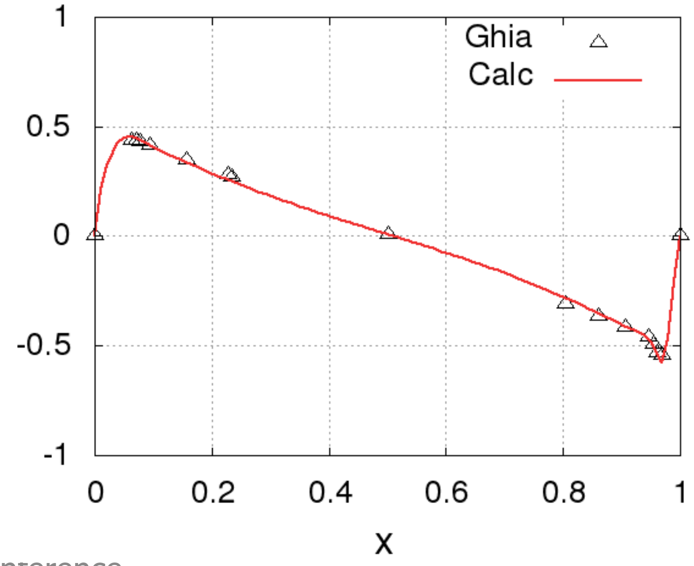
Lid-driven cavity flow (2)

- Solver: modified pimplefoam
- max Co number 50
  - ✓ run in PISO mode
- checker-board in pressure and velocity





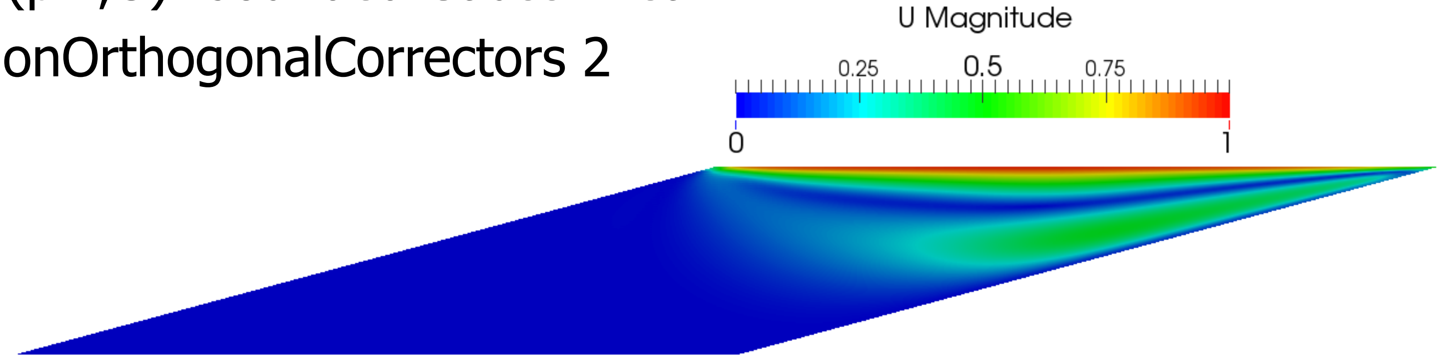


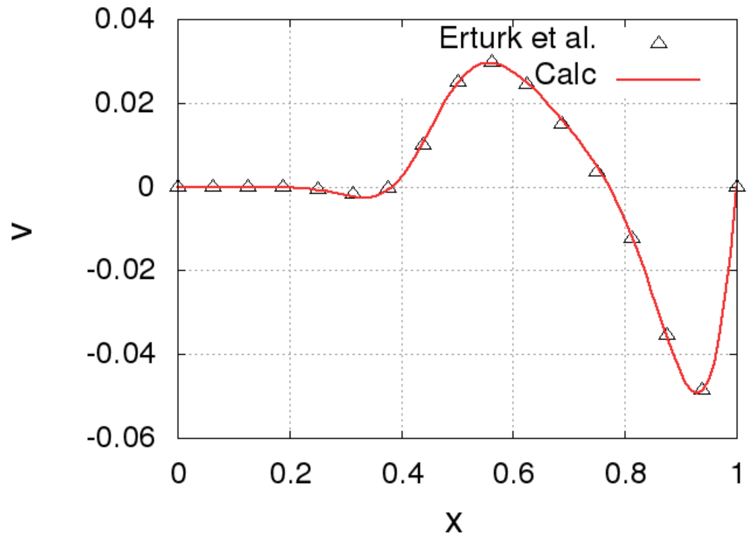


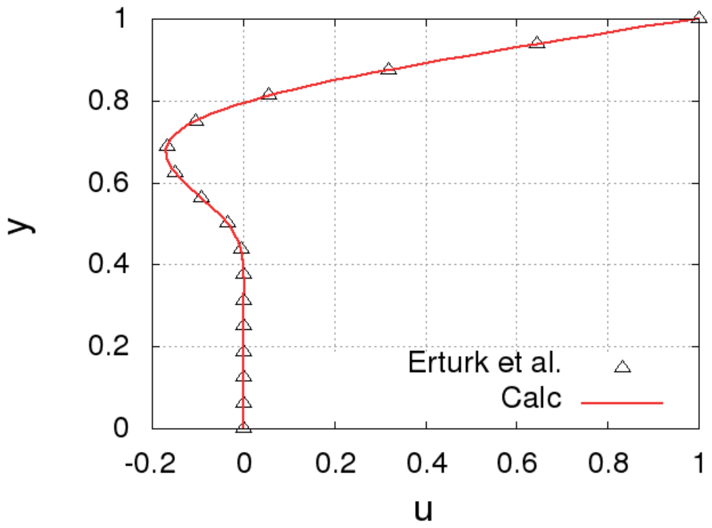
simpleFoam

- **OpenFOAM version**
  - ✓ OpenFOAM dev
- **Solver modification**
  - ✓ additional pressure prediction before solving momentum eq.
- **Test case**
  - ✓ Skewed lid-driven cavity flow
    - cavity skew angle: 15^{o}
    - Re = 1×10^{3}

- **Solver: standard simplefoam**
- **Numerical schemes**
  - ✓ div(phi,U) bounded Gauss linear
  - ✓ nNonOrthogonalCorrectors 2



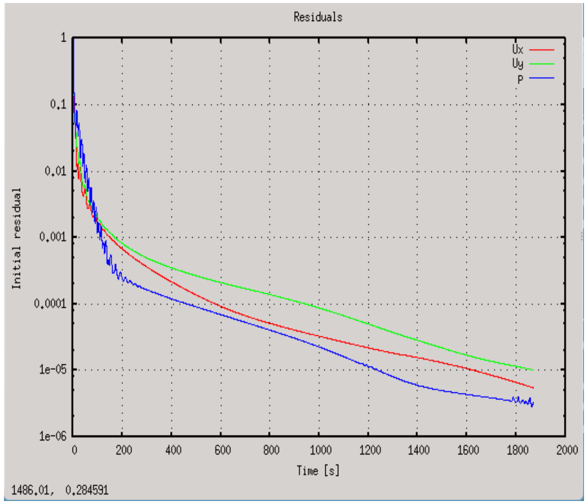


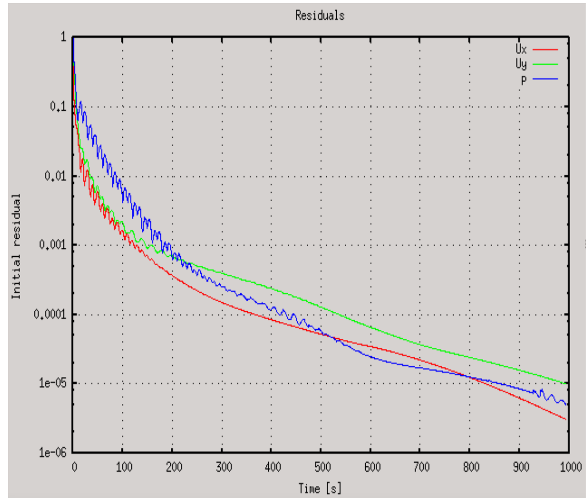


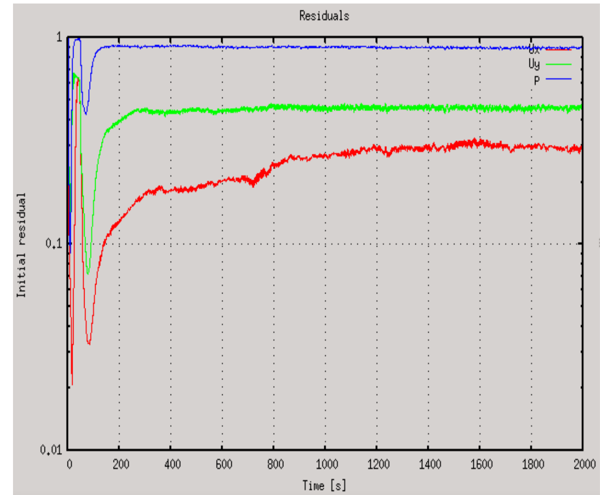
simpleFoam

$\omega_p = 0.5,\ \omega_{\mathbf{U}} = 0.5$ | $\omega_p = 0.3,\ \omega_{\mathbf{U}} = 0.7$ | $\omega_p = 0.7,\ \omega_{\mathbf{U}} = 0.3$

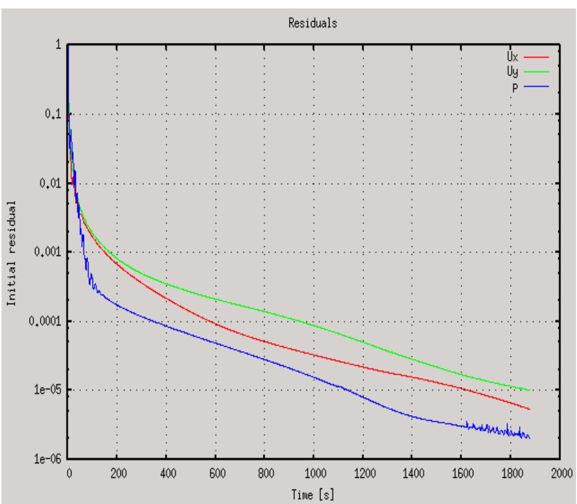
standard

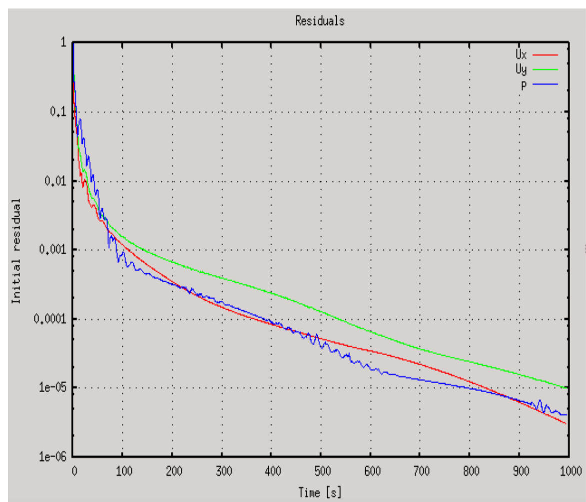


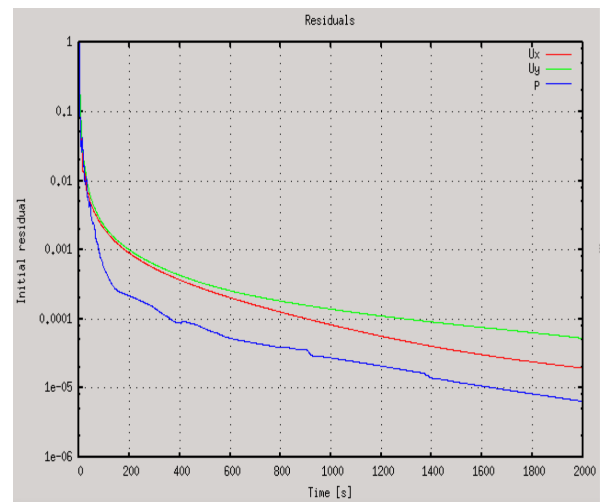




modified





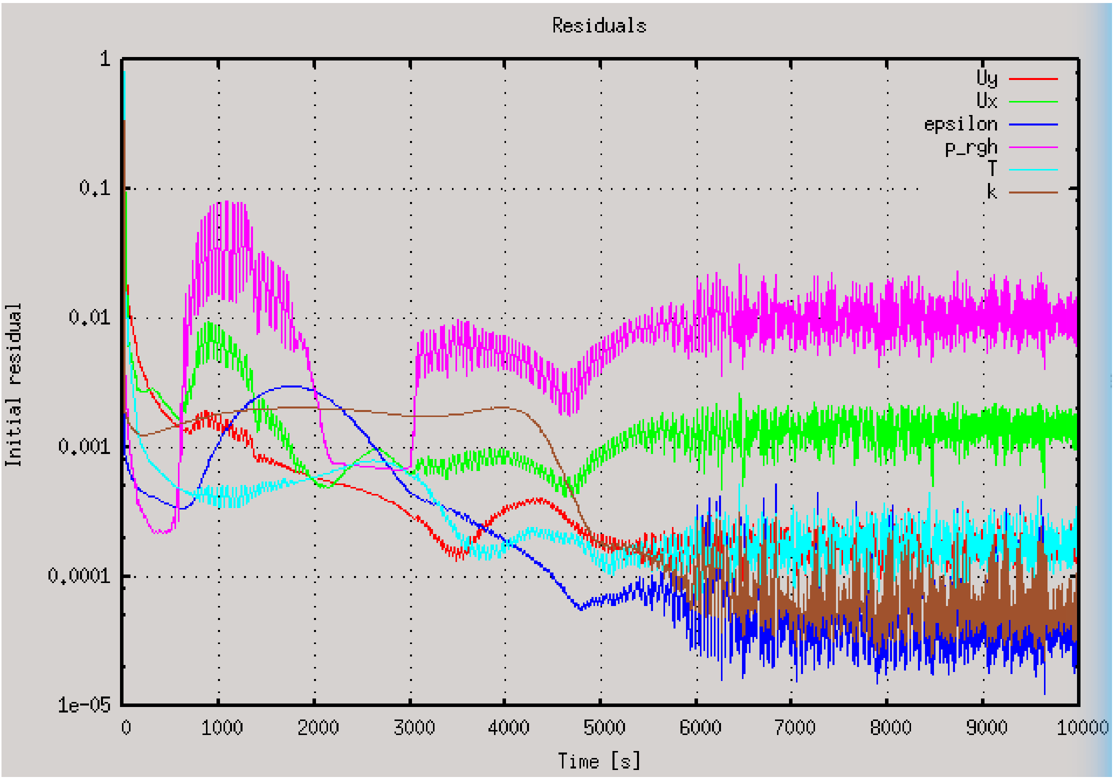


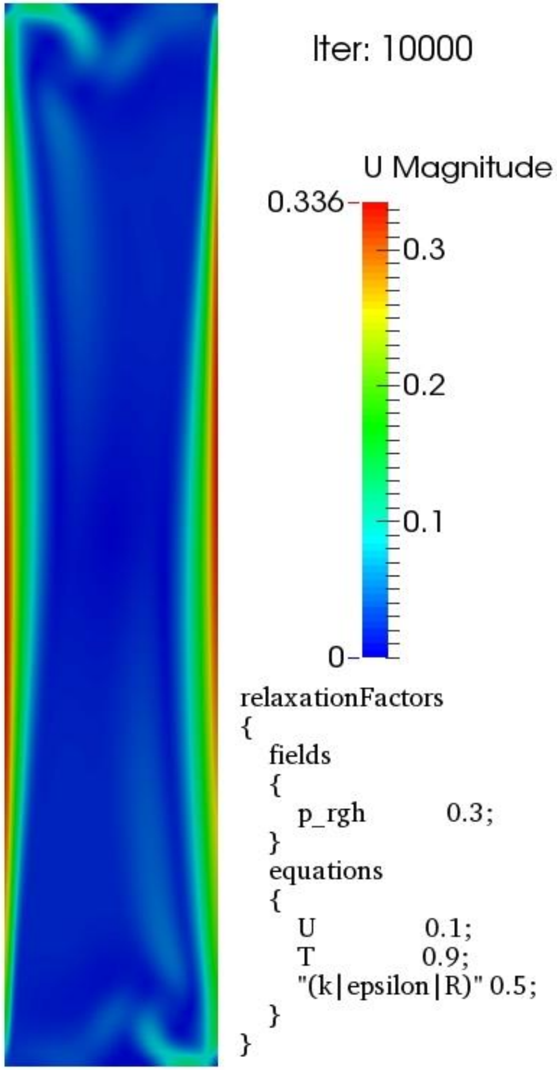
buoyantBoussinesqSimpleFoam

- **OpenFOAM version**
  - ✓ OpenFOAM 2.4
- **Solver modification**
  - ✓ pressure relaxation after velocity update
- **Test case**
  - ✓ King's natural convection problem
    - Ra = 4.3×10^{10}
    - Pr = 0.71

buoyantBoussinesqSimpleFoam

$$\omega_p = 0.3, \quad \omega_{\mathbf{U}} = 0.1$$



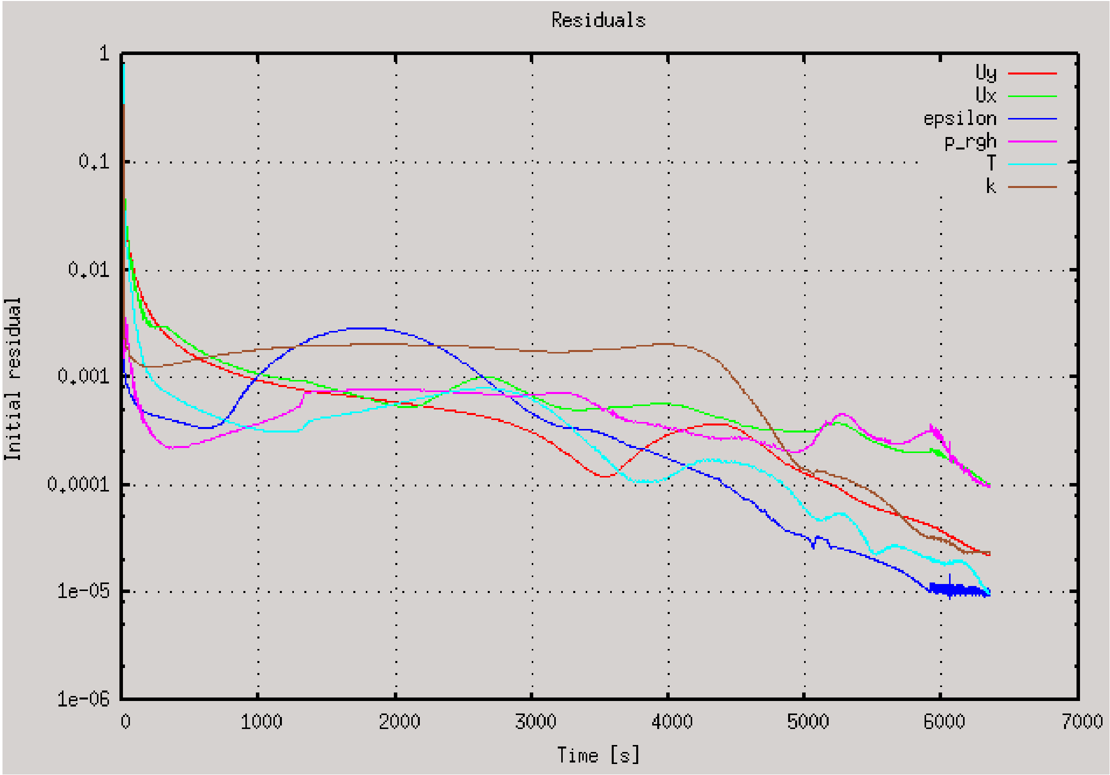


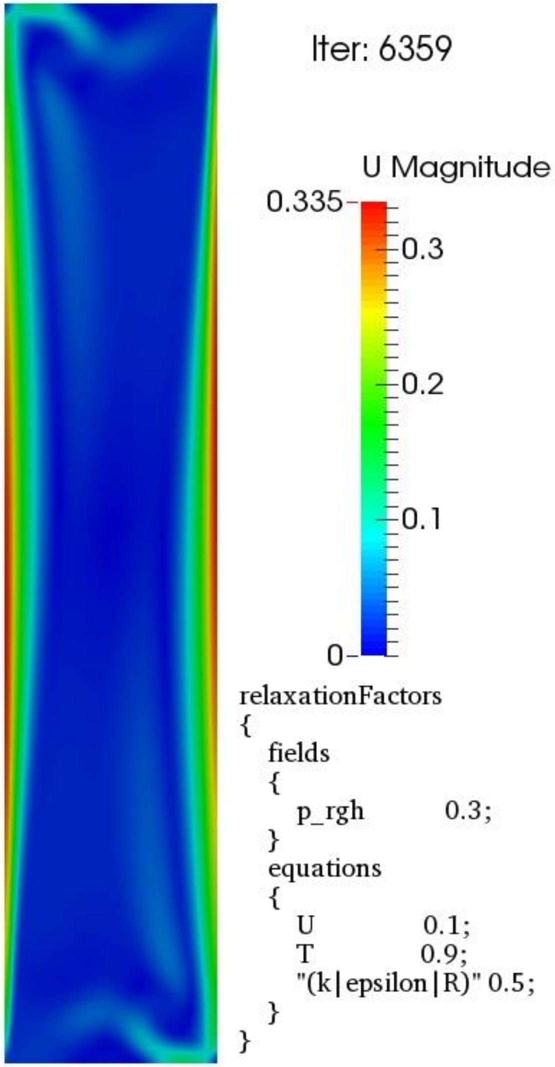

```
relaxationFactors
{
    fields
    {
        p_rgh       0.3;
    }
    equations
    {
        U           0.1;
        T           0.9;
        "(k|epsilon|R)" 0.5;
    }
}
```

Convergence characteristics

modified buoyantBoussinesqSimpleFoam

$$\omega_p = 0.3, \quad \omega_{\mathbf{U}} = 0.1$$



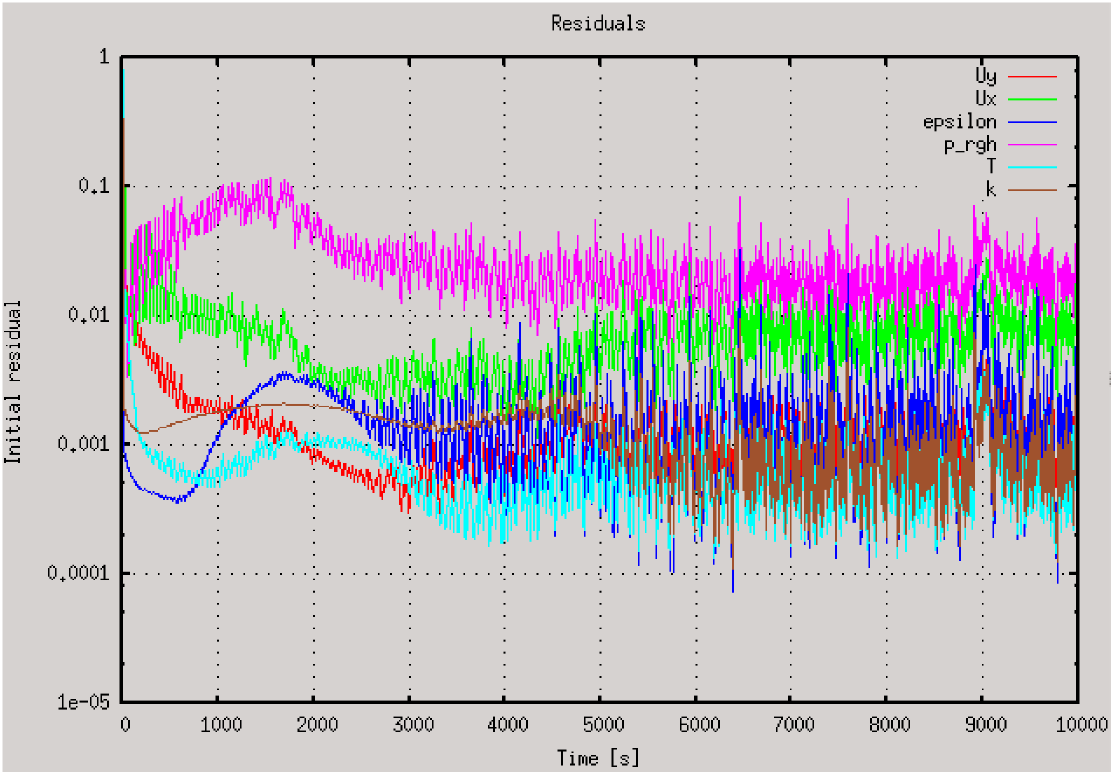


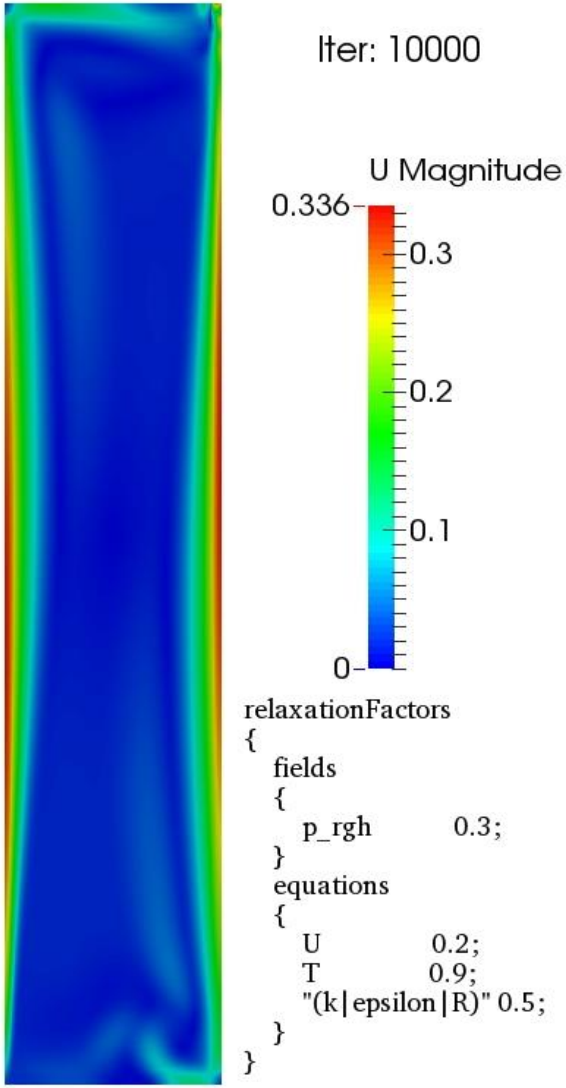

```
relaxationFactors
{
    fields
    {
        p_rgh        0.3;
    }
    equations
    {
        U            0.1;
        T            0.9;
        "(k|epsilon|R)" 0.5;
    }
}
```

Convergence characteristics (4)

buoyantBoussinesqSimpleFoam

$$\omega_p = 0.3, \quad \omega_{\mathbf{U}} = 0.2$$



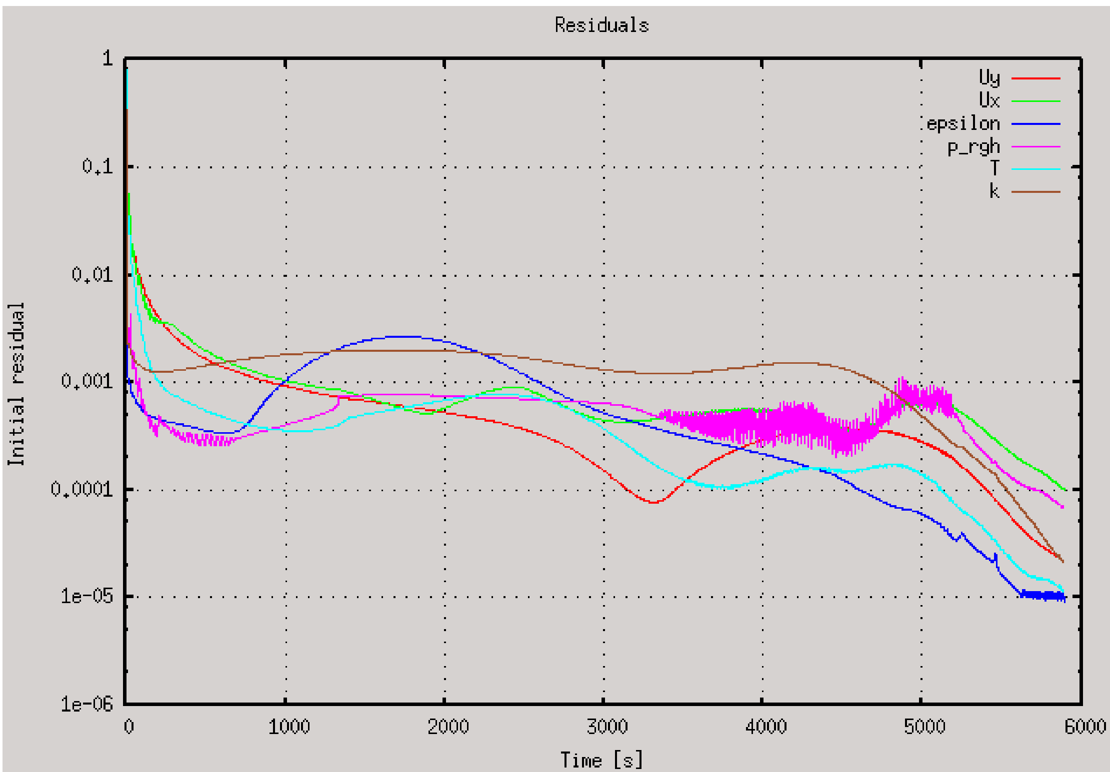



```
relaxationFactors
{
    fields
    {
        p_rgh        0.3;
    }
    equations
    {
        U            0.2;
        T            0.9;
        "(k|epsilon|R)" 0.5;
    }
}
```

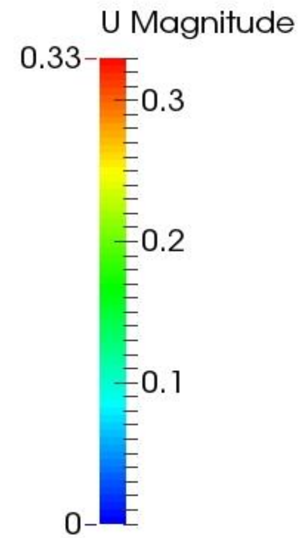
Convergence characteristics (3)

modified buoyantBoussinesqSimpleFoam

$$\omega_p = 0.3, \quad \omega_{\mathbf{U}} = 0.2$$



Iter: 5892

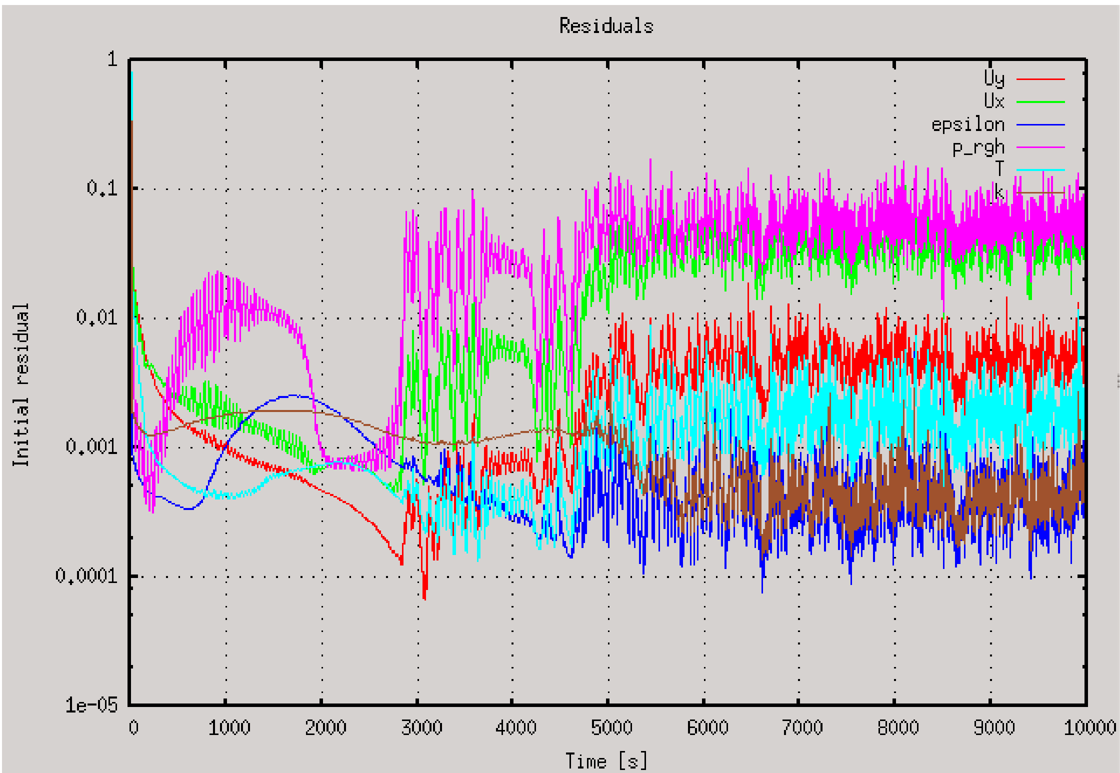


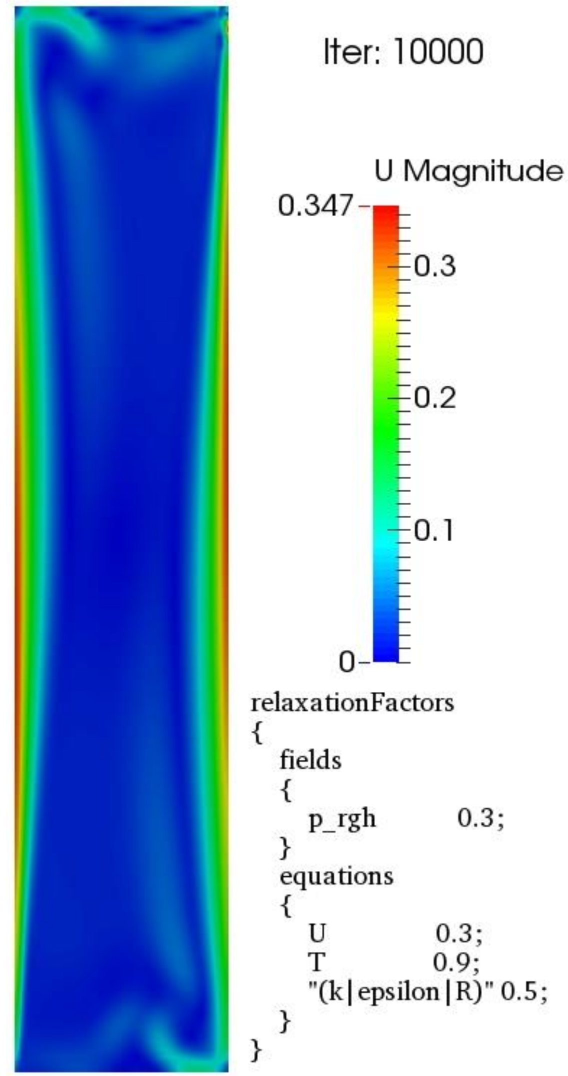

```
relaxationFactors
{
   fields
   {
      p_rgh        0.3;
   }
   equations
   {
      U            0.2;
      T            0.9;
      "(k|epsilon|R)" 0.5;
   }
}
```

Convergence characteristics (5)

modified buoyantBoussinesqSimpleFoam

$$\omega_p = 0.3, \quad \omega_{\mathbf{U}} = 0.3$$



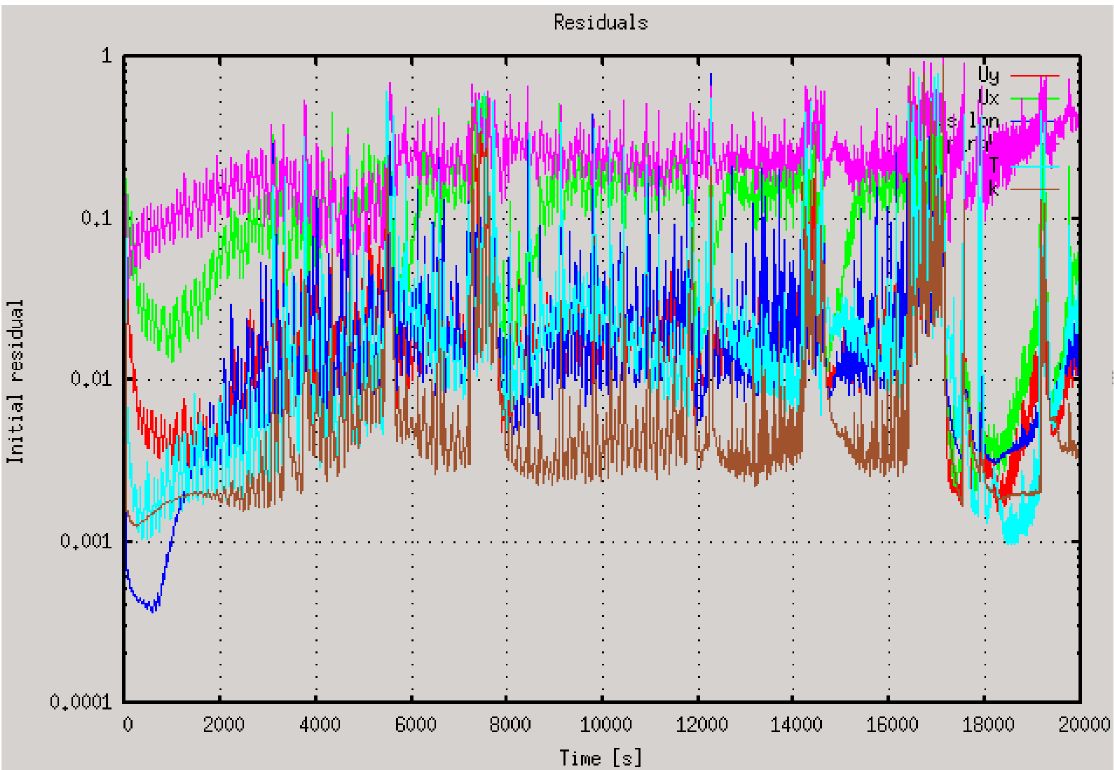


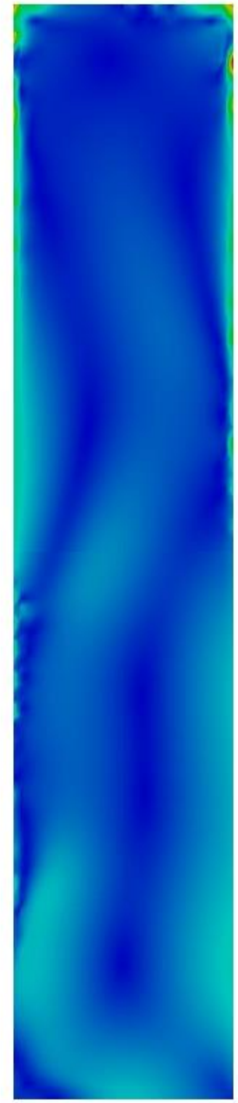

```
relaxationFactors
{
   fields
   {
      p_rgh        0.3;
   }
   equations
   {
      U          0.3;
      T          0.9;
      "(k|epsilon|R)" 0.5;
   }
}
```

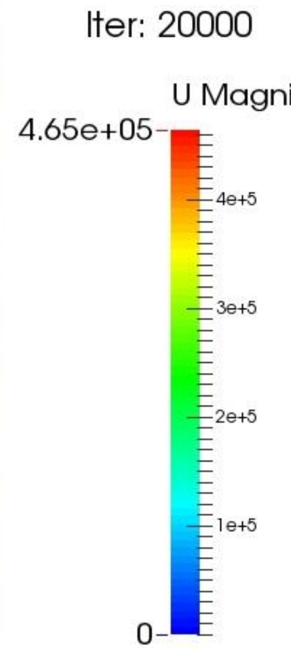
Convergence characteristics (6)

modified buoyantBoussinesqSimpleFoam

$$\omega_p = 0.3, \quad \omega_{\mathbf{U}} = 0.7$$





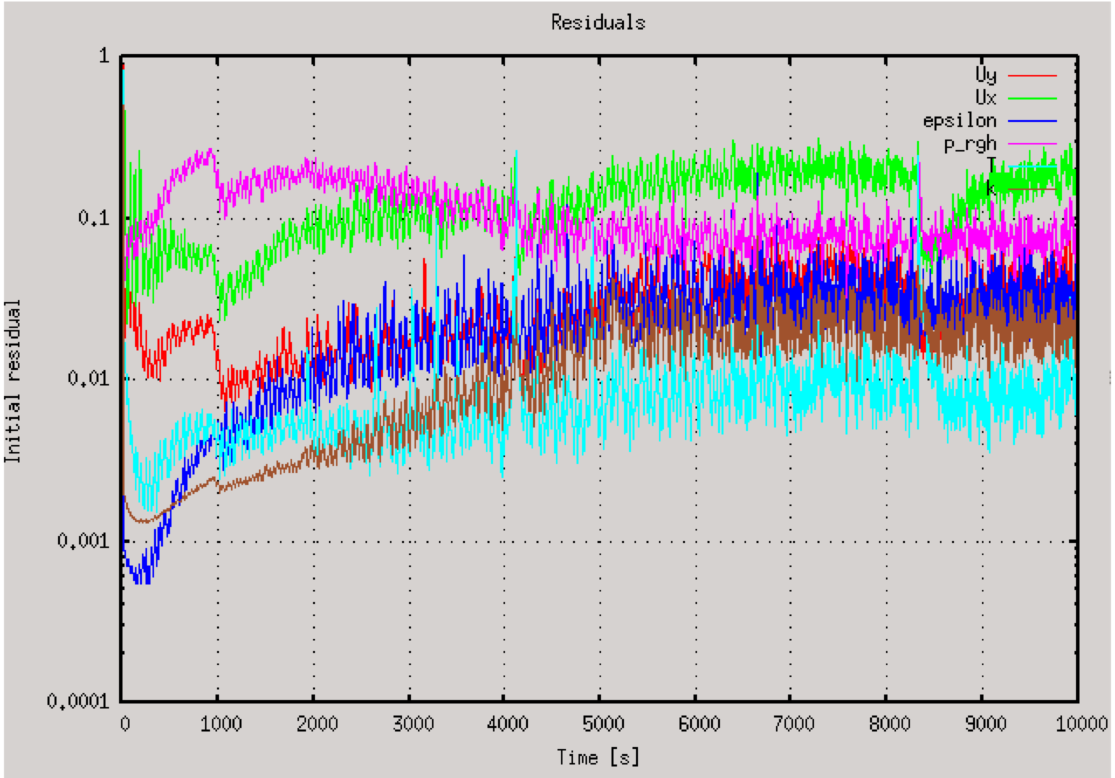


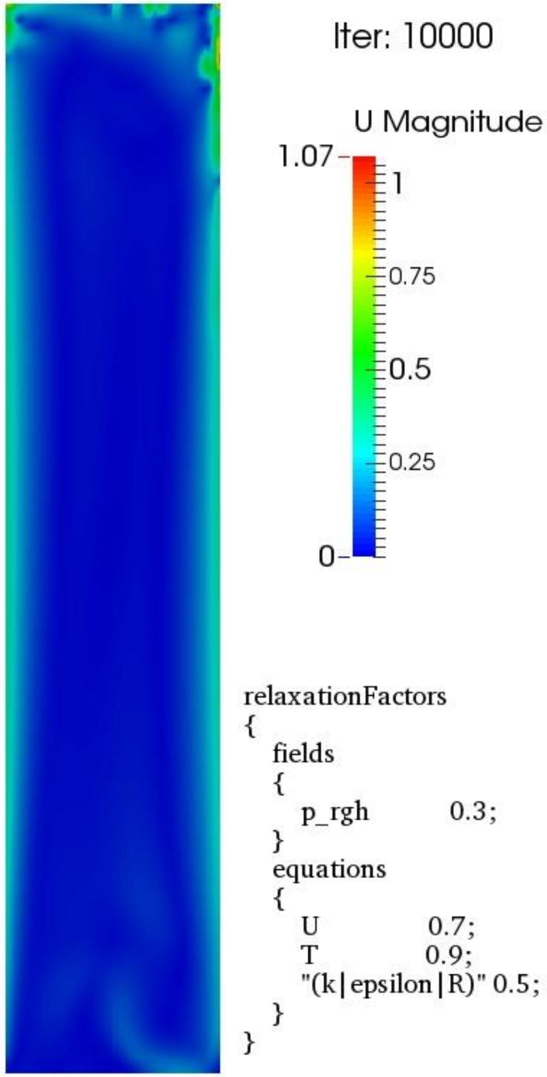

```
relaxationFactors
{
    fields
    {
        p_rgh        0.3;
    }
    equations
    {
        U            0.7;
        T            0.9;
        "(k|epsilon|R)" 0.5
    }
}
```

Convergence characteristics (7)

buoyantBoussinesqSimpleFoam

$$\omega_p = 0.3, \quad \omega_{\mathbf{U}} = 0.7$$

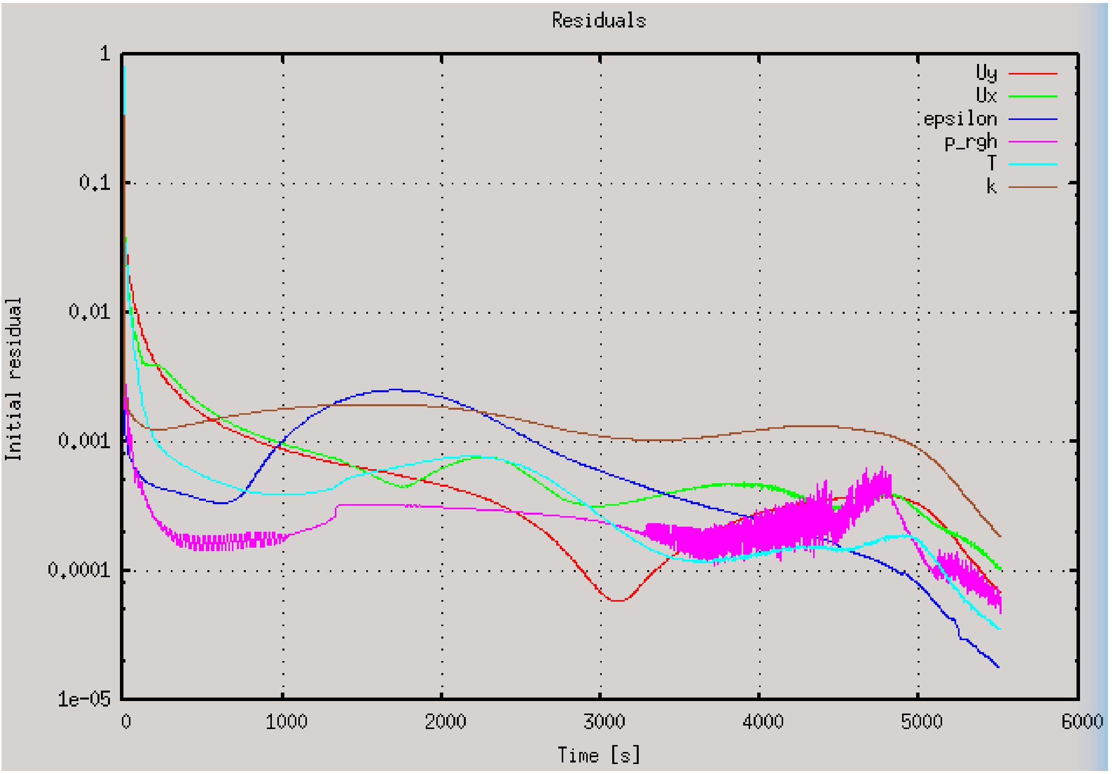


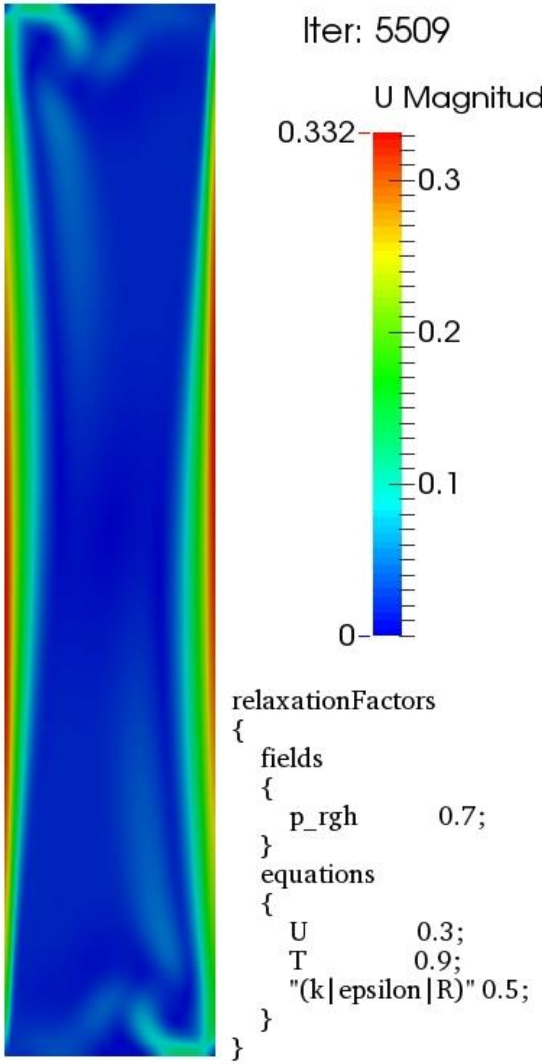



```
relaxationFactors
{
    fields
    {
        p_rgh        0.3;
    }
    equations
    {
        U            0.7;
        T            0.9;
        "(k|epsilon|R)" 0.5;
    }
}
```

modified buoyantBoussinesqSimpleFoam

$$\omega_p = 0.7\,,\quad \omega_{\mathbf{U}} = 0.3$$



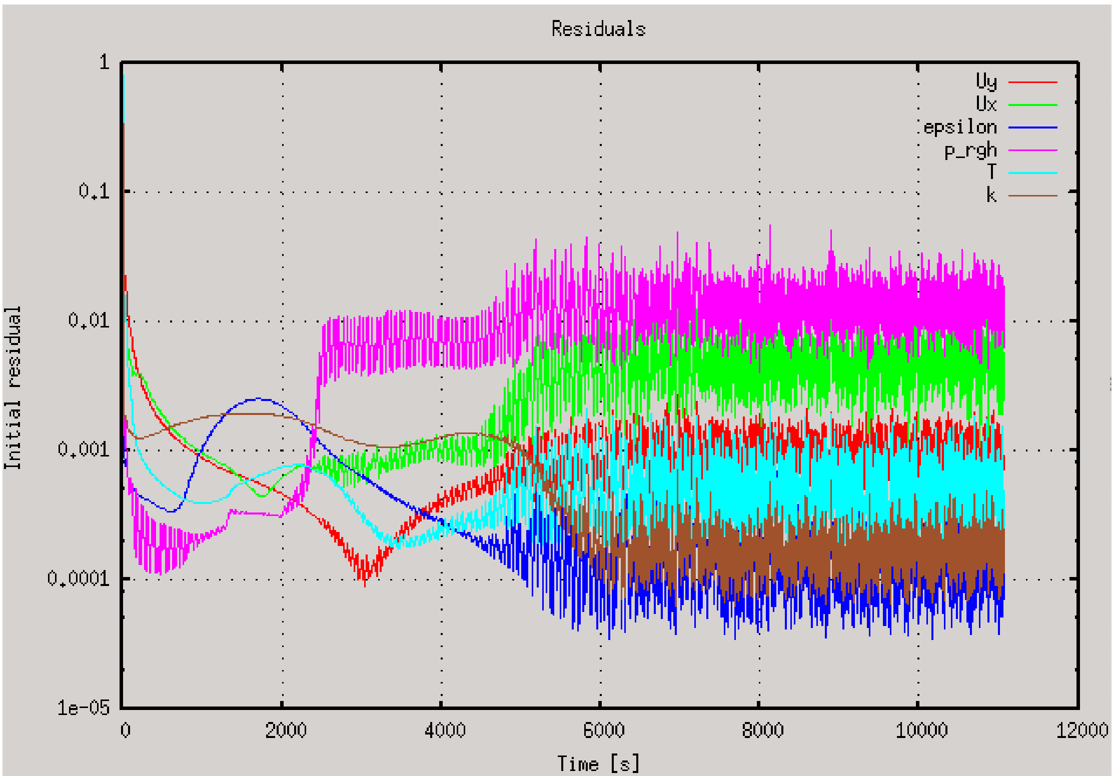


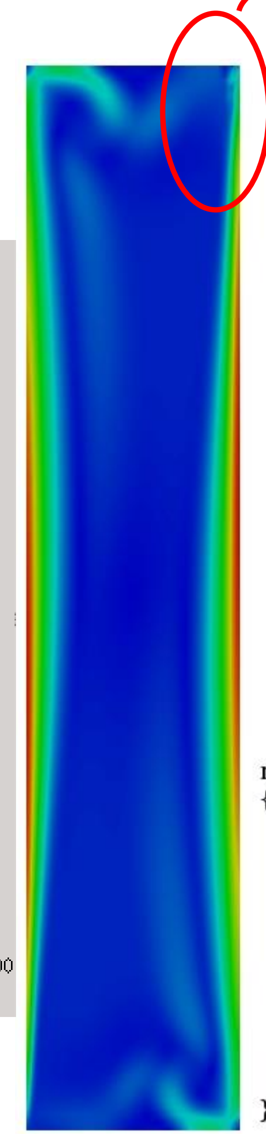

```
relaxationFactors
{
    fields
    {
        p_rgh          0.7;
    }
    equations
    {
        U              0.3;
        T              0.9;
        "(k|epsilon|R)" 0.5;
    }
}
```

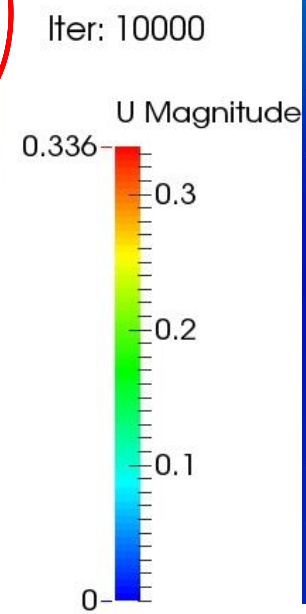
Convergence characteristics (9)

buoyantBoussinesqSimpleFoam

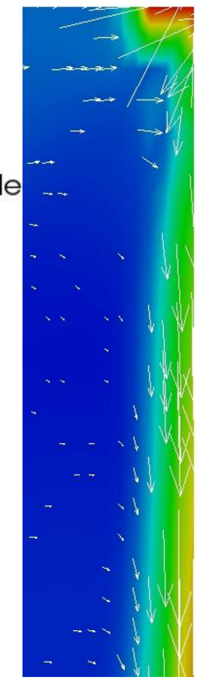
$$\omega_p = 0.7\,,\quad \omega_{\mathbf{U}} = 0.3$$

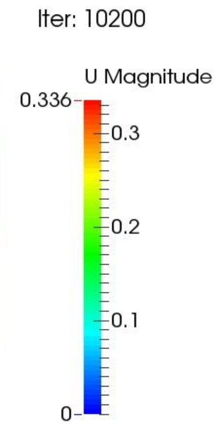







```
relaxationFactors
{
    fields
    {
        p_rgh       0.7;
    }
    equations
    {
        U           0.3;
        T           0.9;
        "(k|epsilon|R)" 0.5;
    }
}
```



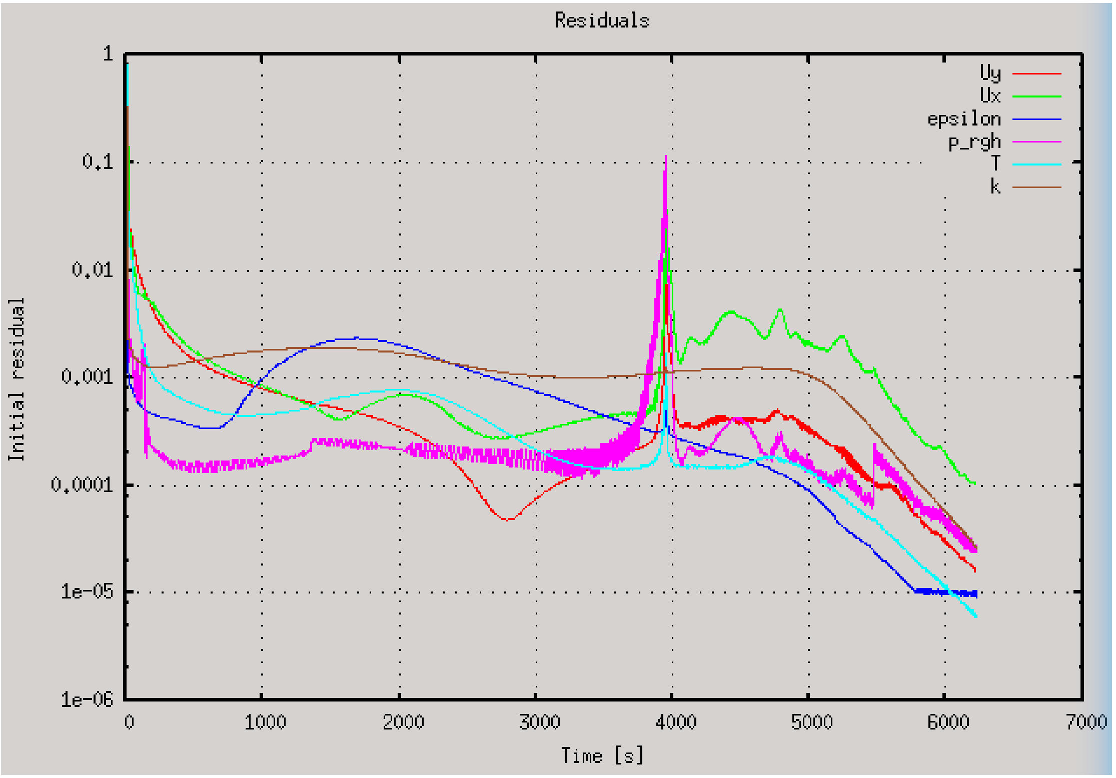


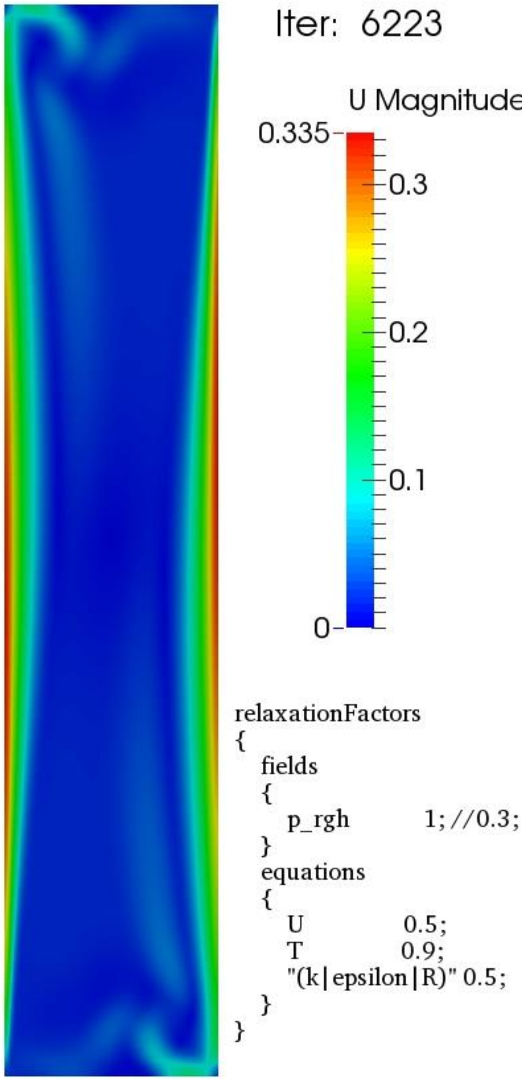

```
relaxationFactors
{
    fields
    {
        p_rgh       0.7;
    }
    equations
    {
        U           0.3;
        T           0.9;
        "(k|epsilon|R)" 0.5;
    }
}
```

Convergence characteristics (8)

modified buoyantBoussinesqSimpleFoam

$$\omega_p = 1, \quad \omega_{\mathbf{U}} = 0.5$$



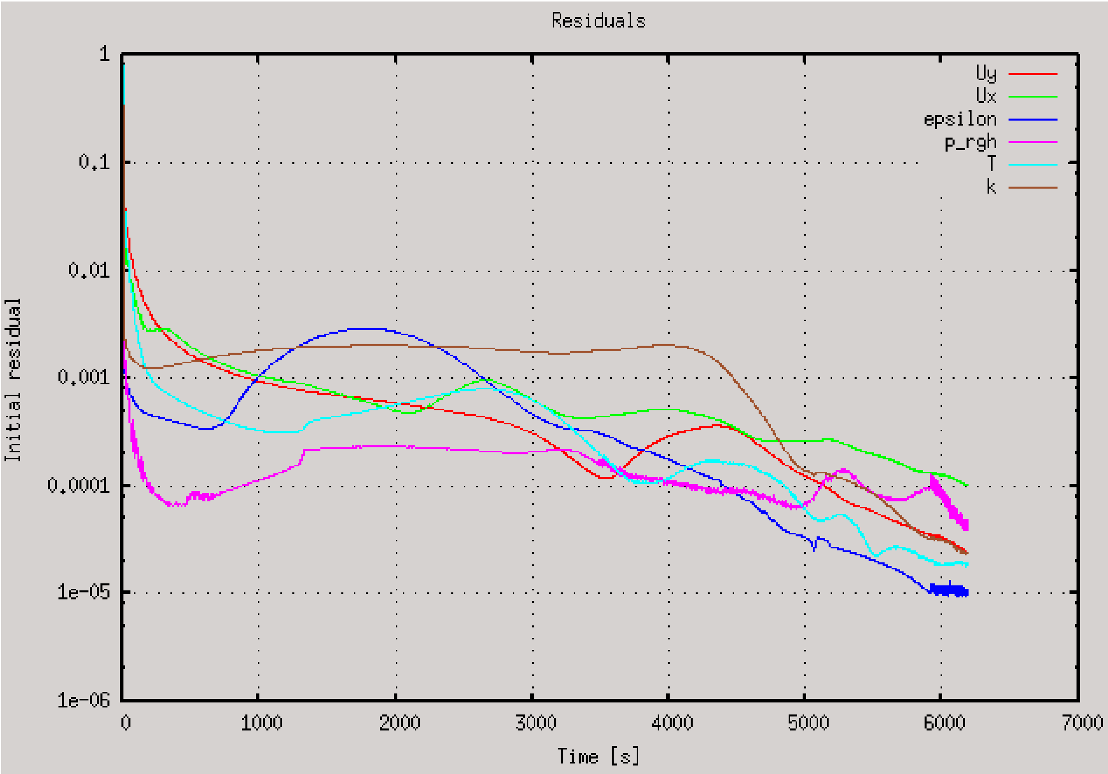


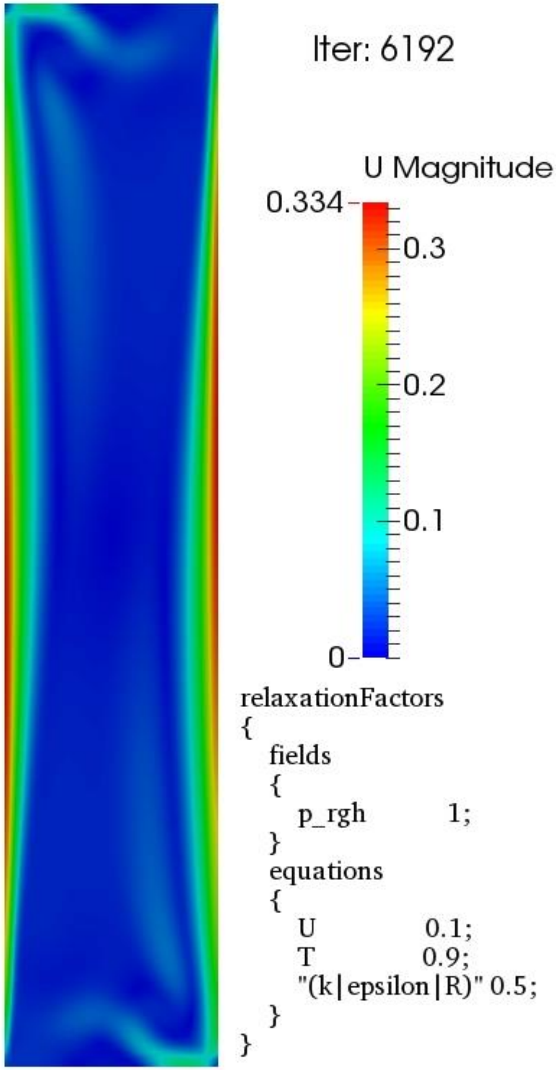

```
relaxationFactors
{
    fields
    {
        p_rgh        1;//0.3;
    }
    equations
    {
        U            0.5;
        T            0.9;
        "(k|epsilon|R)" 0.5;
    }
}
```

Convergence characteristics (9)

modified buoyantBoussinesqSimpleFoam

$$\omega_p = 1, \quad \omega_{\mathbf{U}} = 0.1$$



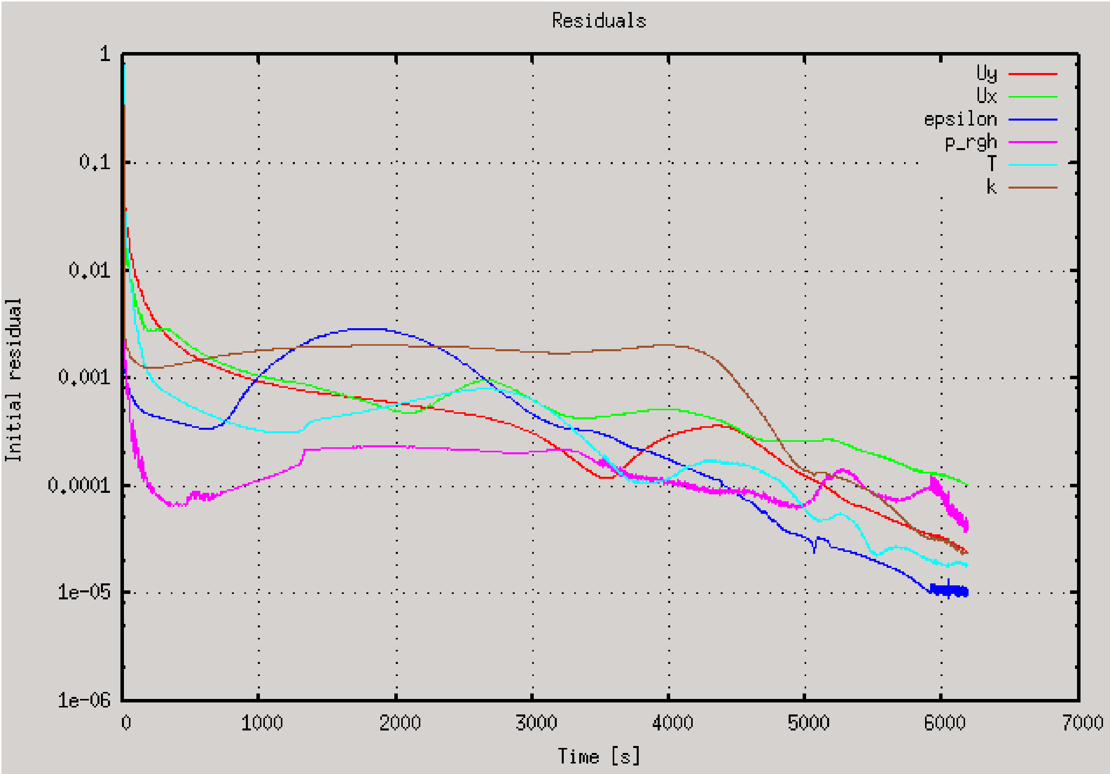


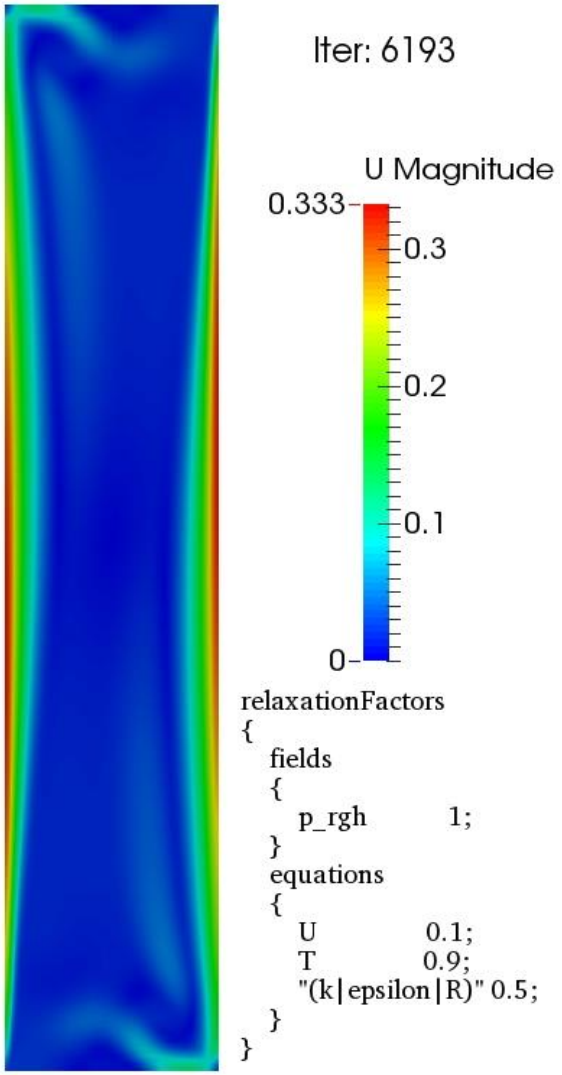

```
relaxationFactors
{
    fields
    {
        p_rgh       1;
    }
    equations
    {
        U           0.1;
        T           0.9;
        "(k|epsilon|R)" 0.5;
    }
}
```

Convergence characteristics (10)

buoyantBoussinesqSimpleFoam

$$\omega_p = 1, \;\; \omega_{\mathbf{U}} = 0.1$$



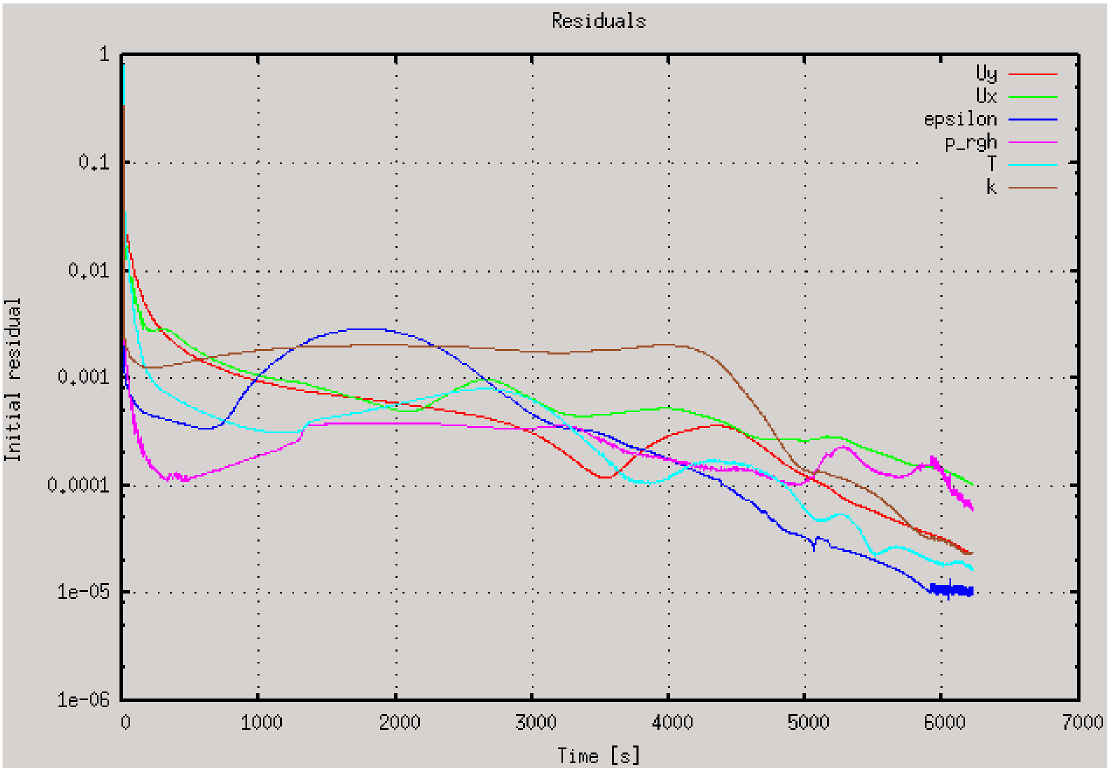


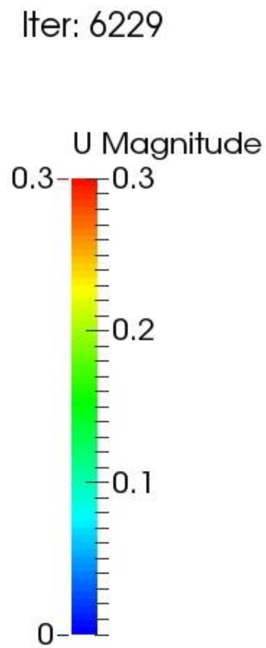

```
relaxationFactors
{
    fields
    {
        p_rgh         1;
    }
    equations
    {
        U             0.1;
        T             0.9;
        "(k|epsilon|R)" 0.5;
    }
}
```

Convergence characteristics (11)

modified buoyantBoussinesqSimpleFoam

$$\omega_p = 0.6, \quad \omega_{\mathrm{U}} = 0.1$$

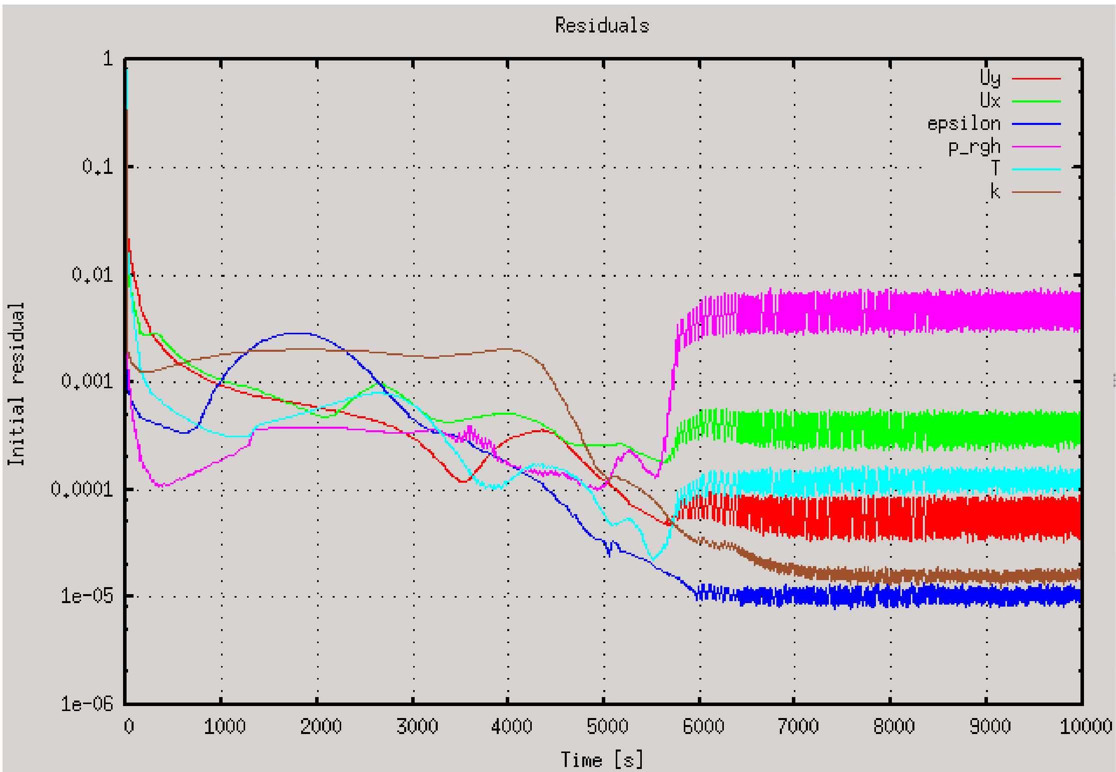


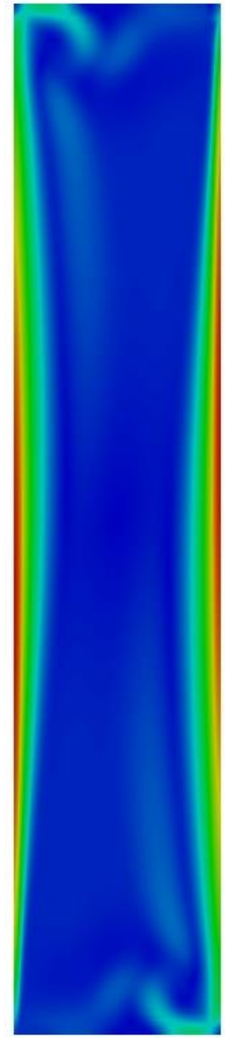


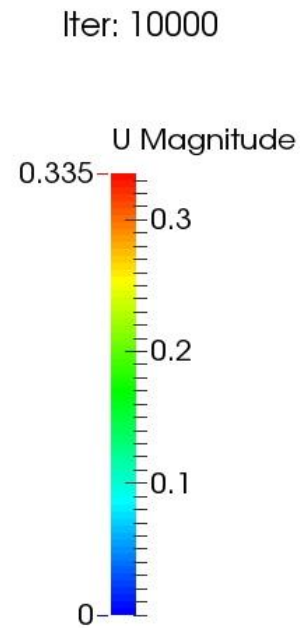

```
relaxationFactors
{
    fields
    {
        p_rgh       0.6;
    }
    equations
    {
        U           0.1;
        T           0.9;
        "(k|epsilon|R)" 0.5;
    }
}
```

buoyantBoussinesqSimpleFoam

$$\omega_p = 0.6\,, \quad \omega_{\mathbf{U}} = 0.1$$





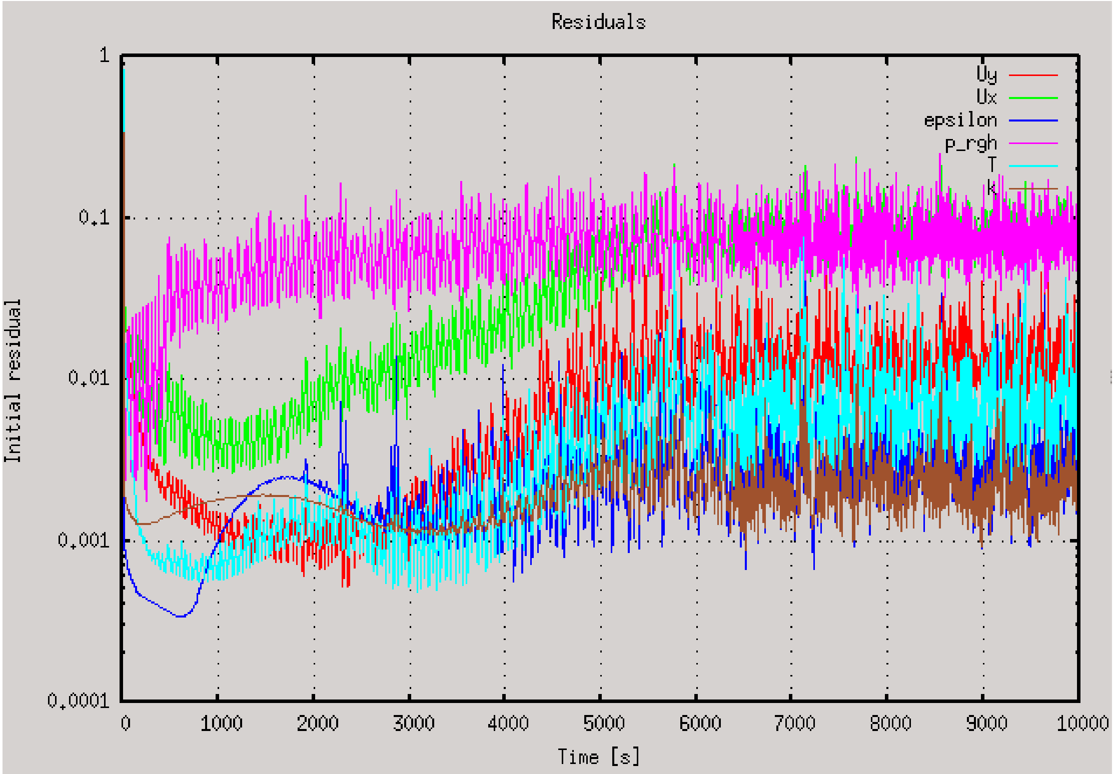


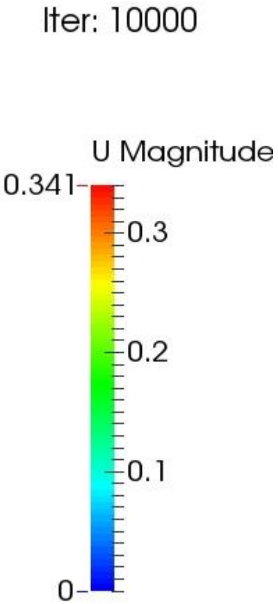

```
relaxationFactors
{
  fields
  {
    p_rgh        0.6;
  }
  equations
  {
    U            0.1;
    T            0.9;
    "(k|epsilon|R)" 0.5;
  }
}
```

Convergence characteristics (13)

modified buoyantBoussinesqSimpleFoam

$$\omega_p = 0.5, \quad \omega_{\mathbf{U}} = 0.5$$



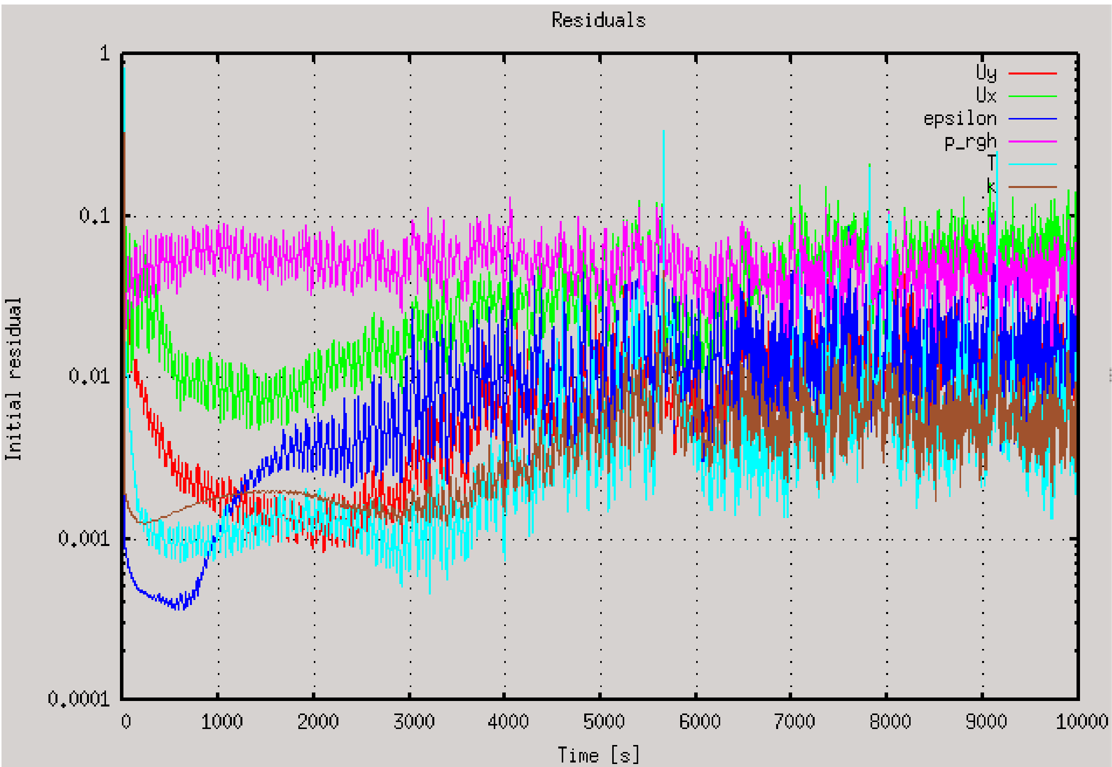


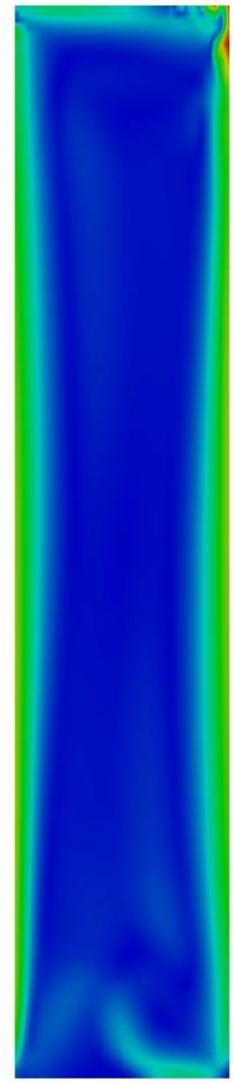

```
relaxationFactors
{
    fields
    {
        p_rgh        0.5;
    }
    equations
    {
        U            0.5;
        T            0.9;
        "(k|epsilon|R)" 0.5;
    }
}
```

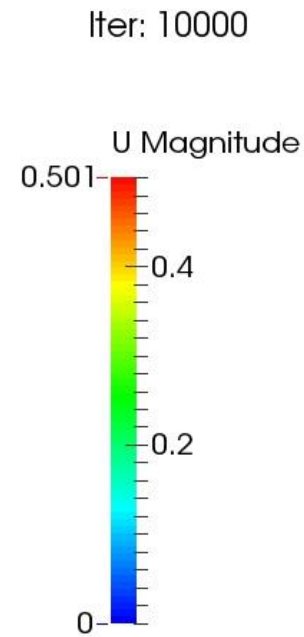
Convergence characteristics (14)

buoyantBoussinesqSimpleFoam

$$\omega_p = 0.5, \quad \omega_{\mathbf{U}} = 0.5$$








```
relaxationFactors
{
    fields
    {
        p_rgh        0.5;
    }
    equations
    {
        U            0.5;
        T            0.9;
        "(k|epsilon|R)" 0.5;
    }
}
```

Summary of Convergence characteristics

modified buoyantBoussinesqSimpleFoam

ω_p	ω_U	Converged (iter.)
1	0.5	6223
0.3	0.7	diverged
0.7	0.3	5509
0.5	0.5	oscillating
0.3	0.3	oscillating
1	0.1	6192
0.6	0.1	6229
0.3	0.1	6359
0.3	0.2	5892

buoyantBoussinesqSimpleFoam

ω_p	ω_U	Converged (iter.)
1	0.5	6223
0.3	0.7	oscillating
0.7	0.3	oscillating
0.5	0.5	oscillating
0.3	0.3	oscillating
1	0.1	6193
0.6	0.1	oscillating
0.3	0.1	oscillating
0.3	0.2	oscillating

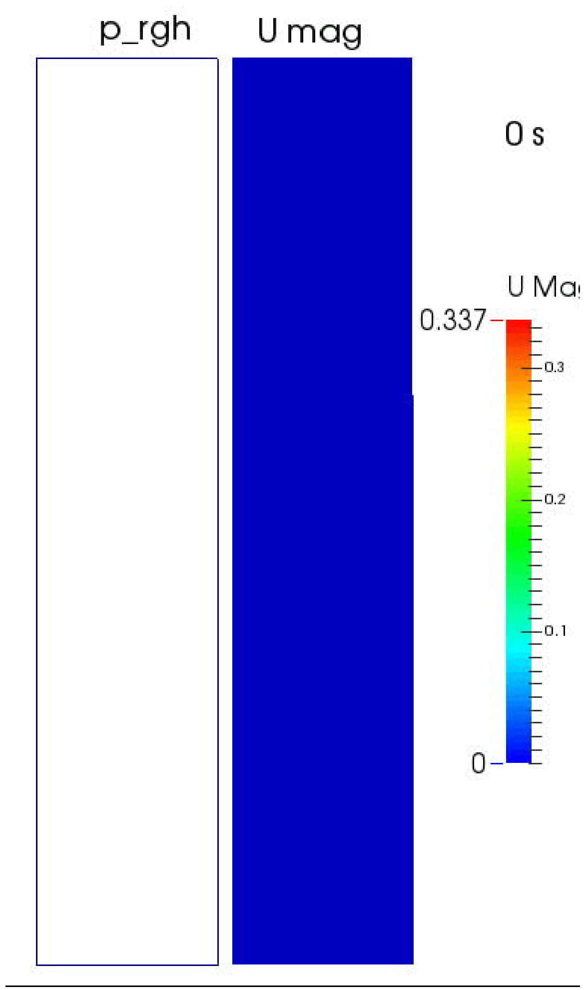
buoyantBoussinesqPimpleFoam

- **OpenFOAM version**
  - ✓ OpenFOAM 2.4
- **Solver modification**
  - ✓ pressure relaxation after velocity update
  - ✓ additional pressure prediction before solving momentum eq.

- **Test case**
  - ✓ King's natural convection problem
    - Ra = 4.3×10^{10}
    - Pr = 0.71

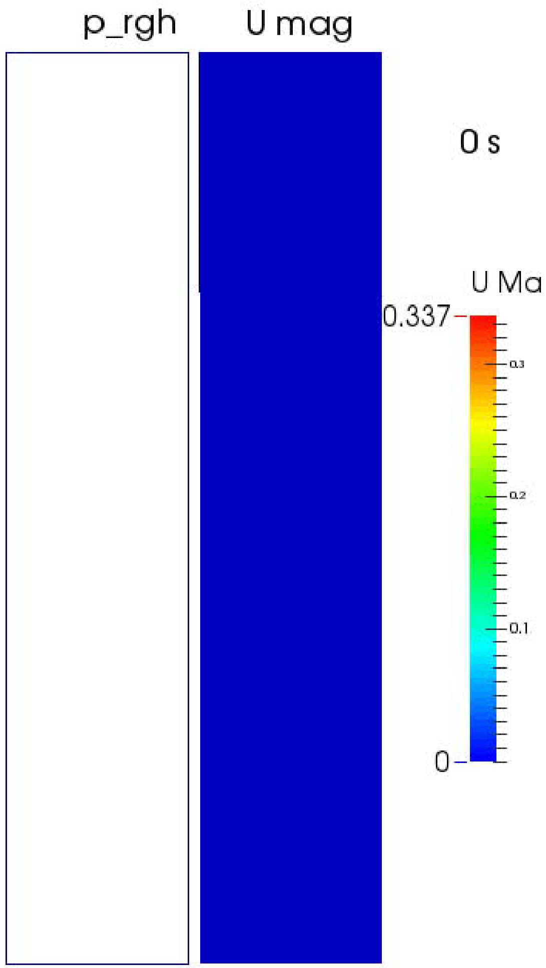
Standard *buoyantBoussinesqPimpleFoam*

- **Solver options**
  - ✓ PISO mode (nOuterCorrectors = 1, nCorrectors = 3) ➔ no relaxation
- **Time step control**
  - ✓ maxDeltaT = 0.1, maxCo = 50
  - ✓ runtime calculated maxCo = 1.85



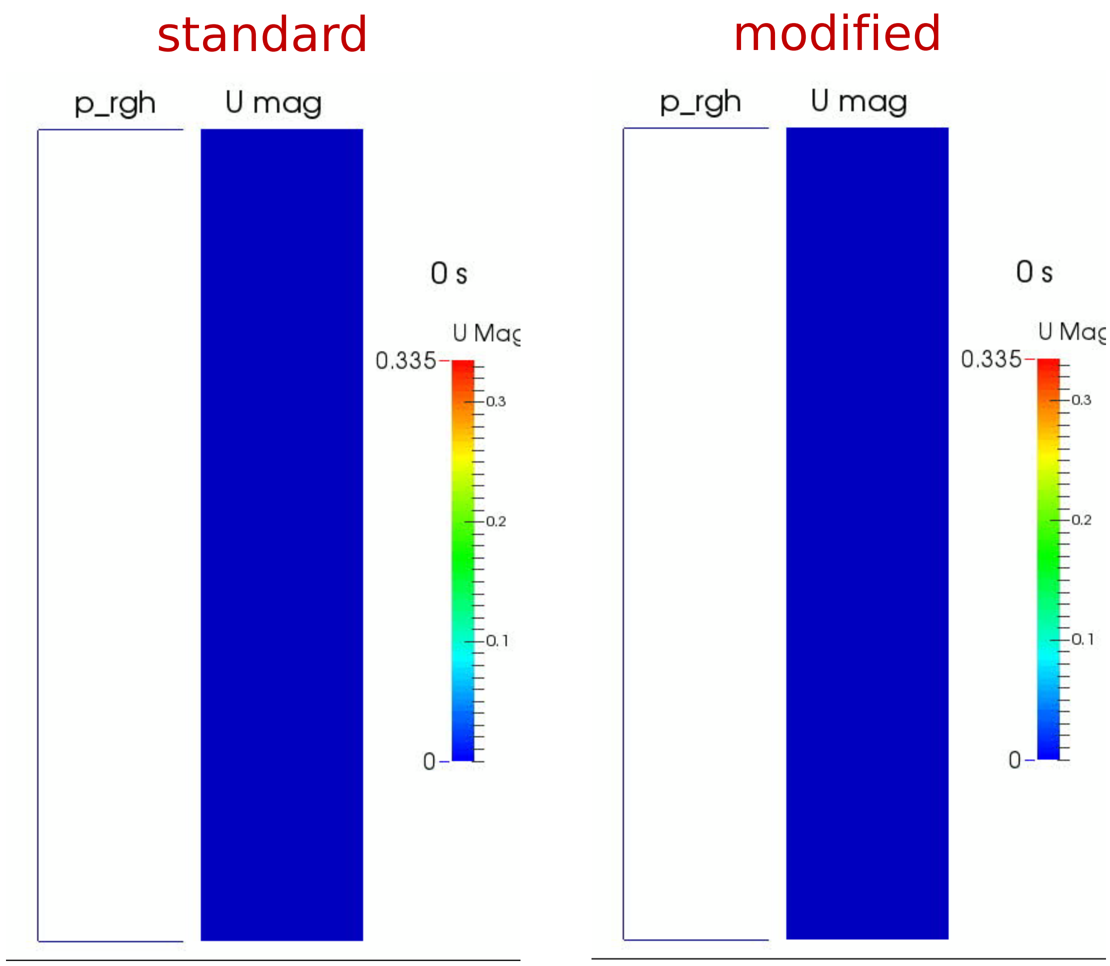
Standard *buoyantBoussinesqPimpleFoam*

- **Solver options**
  - ✓ Pimple mode (nOuterCorrectors = 2, nCorrectors = 3)
- **Relaxation**
  - ✓ p_rgh = 0.5, p_rghFinal = 1
  - ✓ U=0.5, Ufinal=1
- **Time step control**
  - ✓ maxDeltaT = 0.1
- **No effect of relaxation step change**



Modified *buoyantBoussinesqPimpleFoam*

- **Solver options**
  - ✓ Pimple mode (nOuterCorrectors = 2, nCorrectors = 3)
- **Relaxation**
  - ✓ p_rgh = 0.5, p_rghFinal = 1
  - ✓ U=0.5, Ufinal=1
- **Time step control**
  - ✓ maxCo = 10



buoyantPimpleFoam

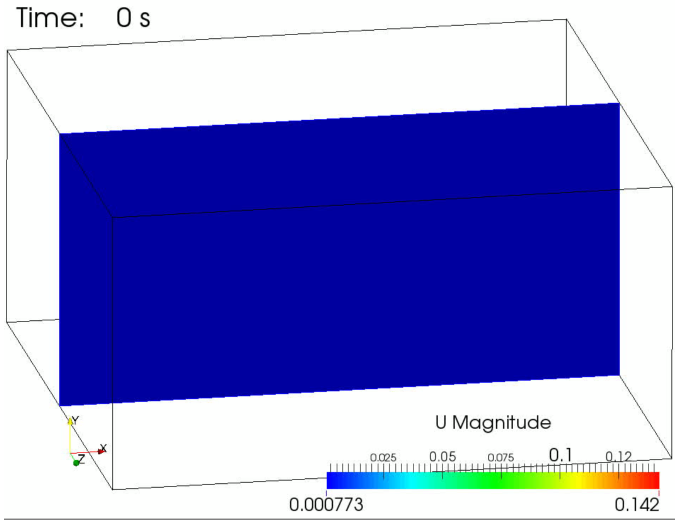
- **OpenFOAM version**
  - ✓ OpenFOAM dev
- **Solver modification**
  - ✓ pressure relaxation after velocity update
  - ✓ additional pressure prediction before solving momentum eq.
- **Test case**
  - ✓ tutorial/heatTransfer/buoyantPimplefoam/hotroom

Standard buoyantPimpleFoam

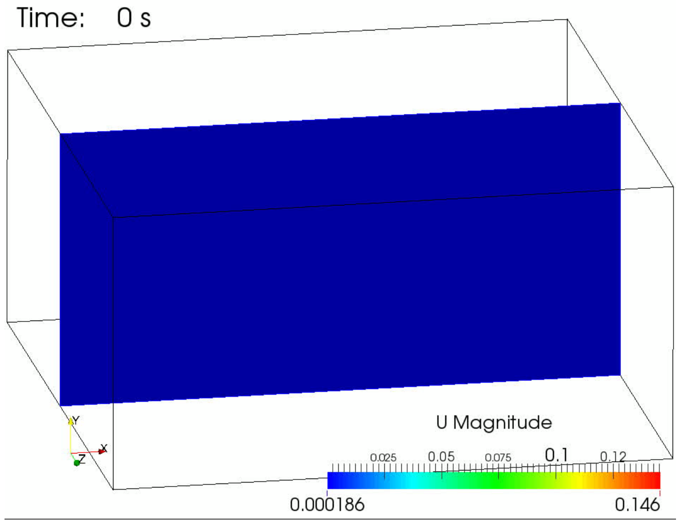
- **Solver options**
  - ✓ PISO mode (nOuterCorrectors = 1, nCorrectors = 2)
- **Convergence**

  only if dt ≤ 2

dt =1, maxCo=0.3



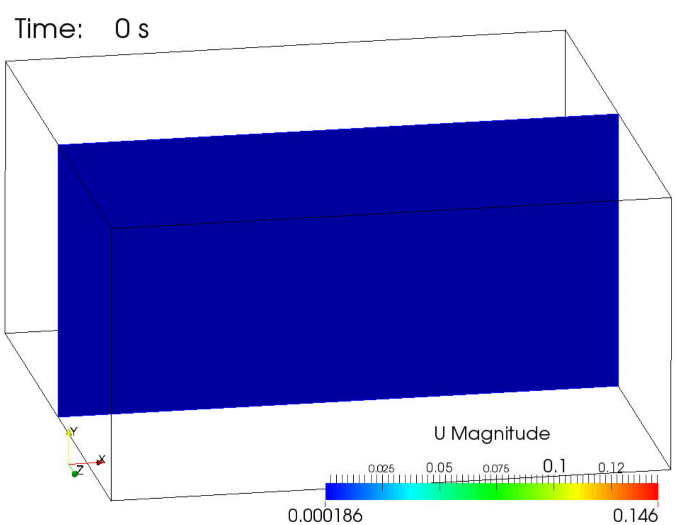
dt =2, maxCo=0.6



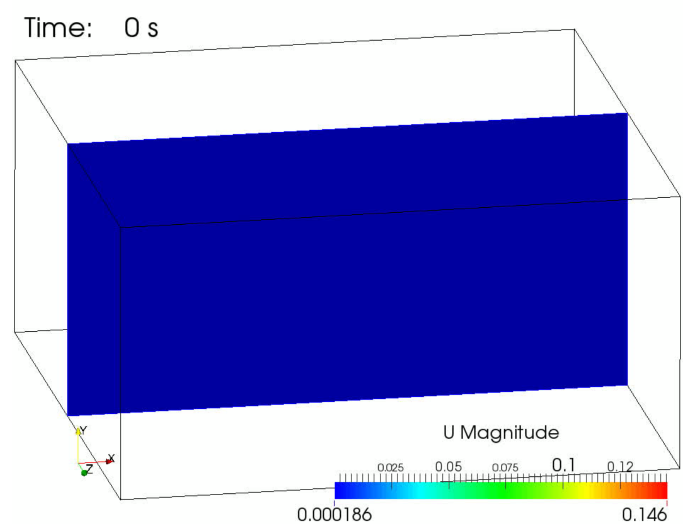
p-relax modified buoyantPimpleFoam

- **Solver options**
  - ✓ PISO mode (nOuterCorrectors = 1, nCorrectors = 2)
- **Convergence**
  - ✓ only if dt ≤ 2
- **In the PISO mode, no relaxation is used**
  - ✓ Convergence characteristics are same as standard solver

dt =1, maxCo=0.3

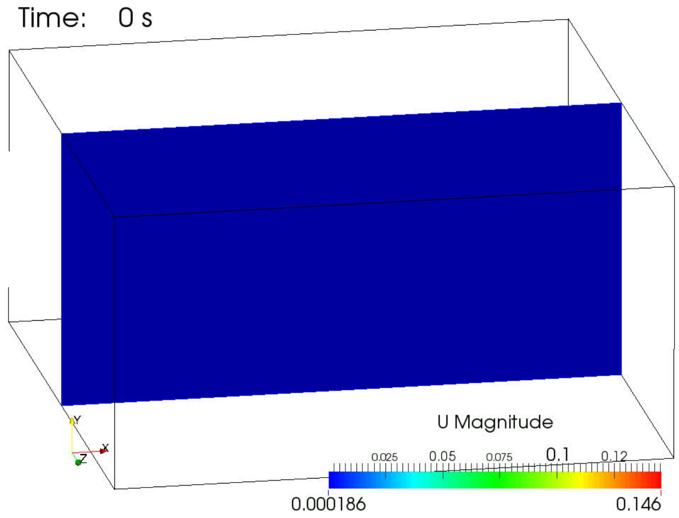


dt =2, maxCo=0.6

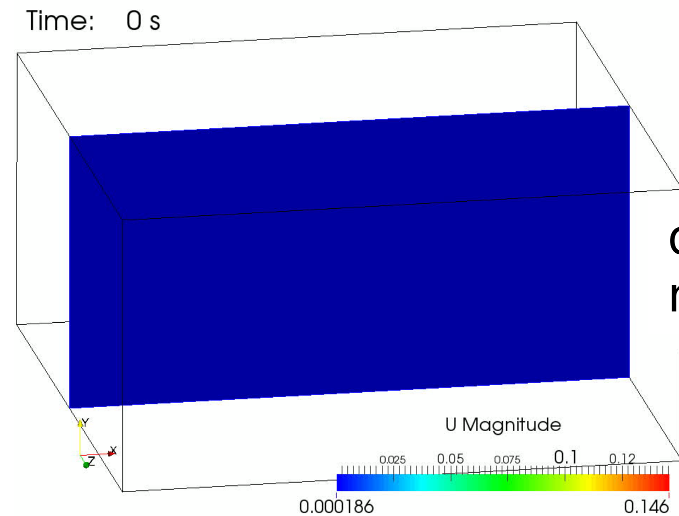


buoyantPimpleFoam with pressure prediction

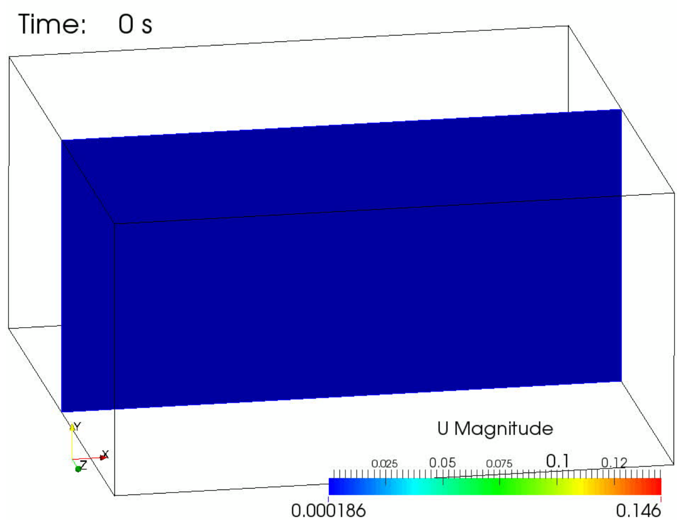
dt =2
maxCo=0.6



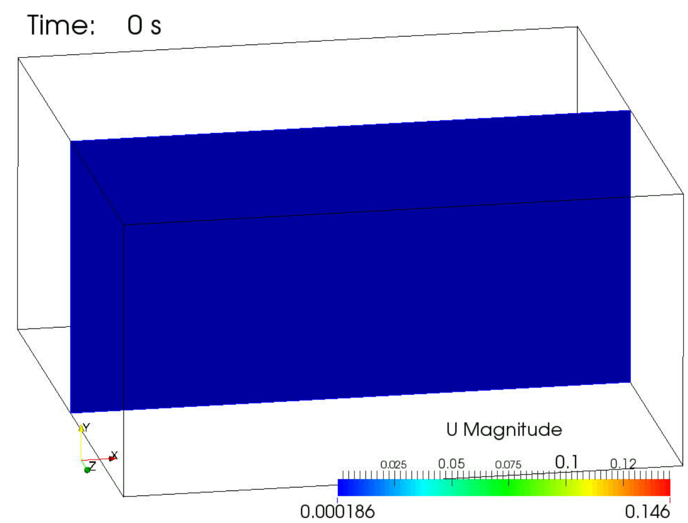
dt =4
maxCo=1.2



dt =10
maxCo=3



dt =20
maxCo=6



Summary

- **Modification of the pressure relaxation step**
  - ✓ It is effective for convergence enhancement of buoyantBoussinesq-SimpleFoam

- **Addition of pressure prediction step**
  - ✓ Increase stability range of relaxation factors in Simple mode and max. Courant number in Piso mode.

- **It is still under evaluation**

- **A critical point to solution divergence**
  - ✓ turbulence model is very critical to the solution convergence

Thank you for your attention